USING SPARSE PARAMETRIZATION OF DEFORMATION FIELDS AS MEANS TO CLASSIFICATION

by

Nishith Tirpankar

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computing

School of Computing

The University of Utah

May 2013

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Nishith Tirpankar**

has been approved by the following supervisory committee members:

Guido Gerig	, Chair	**11/28/2012** Date Approved
Sarang Joshi	, Member	**11/28/2012** Date Approved
Tom Fletcher	, Member	**11/28/2012** Date Approved
Stanley Durrleman	, Member	**12/10/2012** Date Approved

and by **Alan Davis** , Chair of

the Department of **School of Computing**

and by Donna M. White, Interim Dean of The Graduate School.

ABSTRACT

Large Deformation Diffeomorphic Metric Mapping is a powerful technique which has been used to quantify variations in anatomical structures in medical images. This allows us to compare various images within and across a populations of classes using the underlying deformation field which maps each image with the representative images of the class. The deformation field can be described by a low-dimensional control point parameterization. We investigate the potential of this low-dimensional parameterization in classification and study the effect of the underlying classifier parameters on the classification accuracy.

To Nikhil, Anshul, Mark, and Purba for being such a great support.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The problem of statistics on high-dimensional image data

Image data are intrinsically high dimensional. In a dataset of such images, the possible variability is on the order of the size of the images. In many finite databases, the underlying variability can be described on a much lower dimensionality. The variability is constrained by the characteristics of the dataset itself. Medical image datasets tend to have smooth spatial variations characteristic of the fact that locally, pixels do not move independently. Handwritten image datasets have variations only along the contours of the writing. Surveillance datasets have variations along principal paths of travel, restricted to certain parts of the image. In order to perform statistical analysis on such data, it is important to parametrize the variability in the dataset efficiently, in order to reduce the dimensionality while maintaining the information desired. Understanding the constraints posed by the variational characteristics of the data helps in this parametrization. Also, in order to perform statistics, particularly binary and multiclass classification, we will use a registration framework that maintains the information for the task while reducing the size of the descriptor.

1.2 Velocity field as a feature

Landmark matching-based image registration between two images using the technique of large deformation diffeomorphic metric matching [8] finds a smooth velocity field that warps one image to minimize the smoothness constraint and L^2 norm between them. The smoothness constraint is characteristic of the dataset we have chosen. This underlying velocity field is a representative of the variation between the two images.

The velocity field is described completely by the momenta vectors at landmark positions. The variation required to warp one image to another can thus be encoded in the momentum vectors at landmarks. Since the velocity field gives us a measure of the change required to warp one image to another, it can be used as a feature to measure variation between images.

Please note that we will be using the small deformation approximation to the large deformation framework. This implies that the estimated deformations may not be diffeomorphic. This is done in order to simplify the process and reduce simulation times for gradient descent. The small deformation framework can easily be switched for the large deformation diffeomorphic model easily.

1.2.1 Datasets used

We have mainly worked with two datasets. The first is a set of synthetic Snowman data. Some examples of the dataset have been shown in the Figure 1.1.

This dataset consists of 4 images alone and served as a good working dataset for testing the methods initially.

The second dataset which has been used for most of our work is the zip digits handwritten database taken from [6]. It consists of two repositories for training and test. The dataset consists of normalized handwritten digits automatically scanned from the envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; they have been deslanted and size normalized, resulting in 16x16 gray-scale images. There are 7291 images in the training dataset and 2007 images in the test dataset. Table 1.1 shows the digits per class in each of the datasets.

Each line in the data file consists of the digit id (0-9) followed by 256 gray-scale values. Each gray-scale value lies between -1 and 1. The dataset is available at [4].

Figure 1.2 shows a few images of the digits from the training dataset. As it can be seen, the digits are size normalized to fit the 16×16 boundaries and are also deslanted and centered.

The training dataset is considerably large and hence can be assumed to contain most of the variation that people introduce while writing the digits. It is certainly an exhaustive database to train from.

In order to test the classifier, the additional test database has been provided. As seen in Figure 1.3, the images from this dataset are considerably difficult to classify.

Consider the image of the handwritten 2. The loop at the base of the 2 makes it considerably different from the handwritten 2 without the loop. The digit 4 looks similar to a 9. In fact, as a tip from the dataset providers, the test dataset is notoriously difficult and an error rate of 2.5% is excellent.

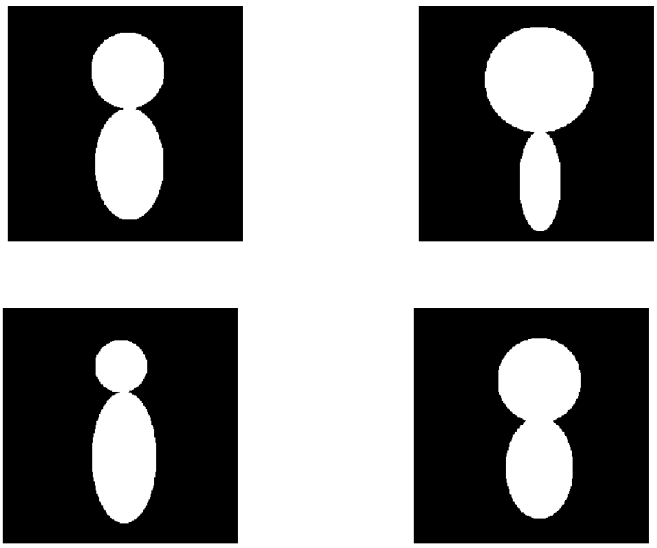


Figure 1.1. Synthetic snowman dataset.

Table 1.1. Distribution of ZIP codes digits.

	0	1	2	3	4	5	6	7	8	9	Total
Train	1194	1005	731	658	652	556	664	645	542	644	7291
Test	359	264	198	166	200	160	170	147	166	177	2007

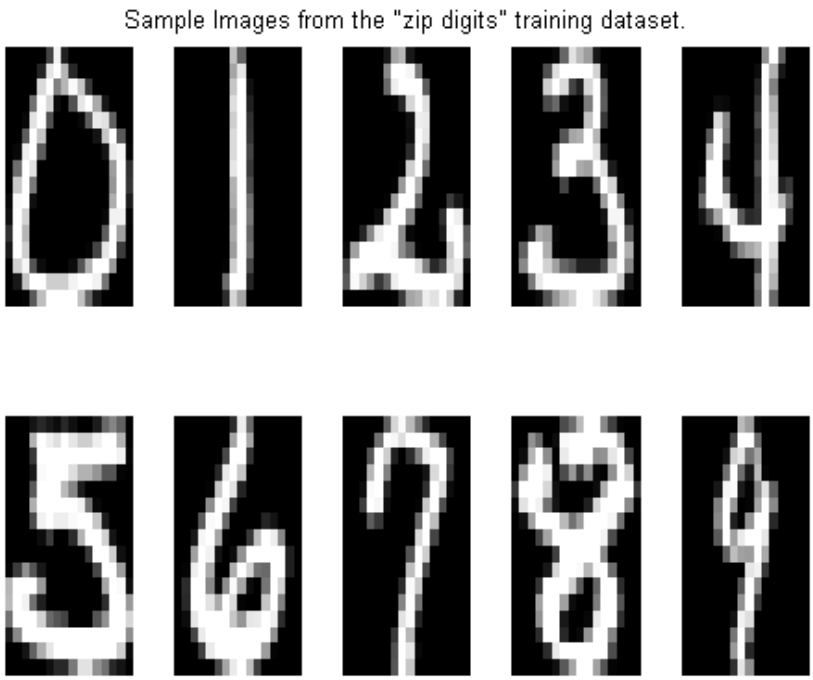


Figure 1.2. Sample images from ZIP code digits training dataset.

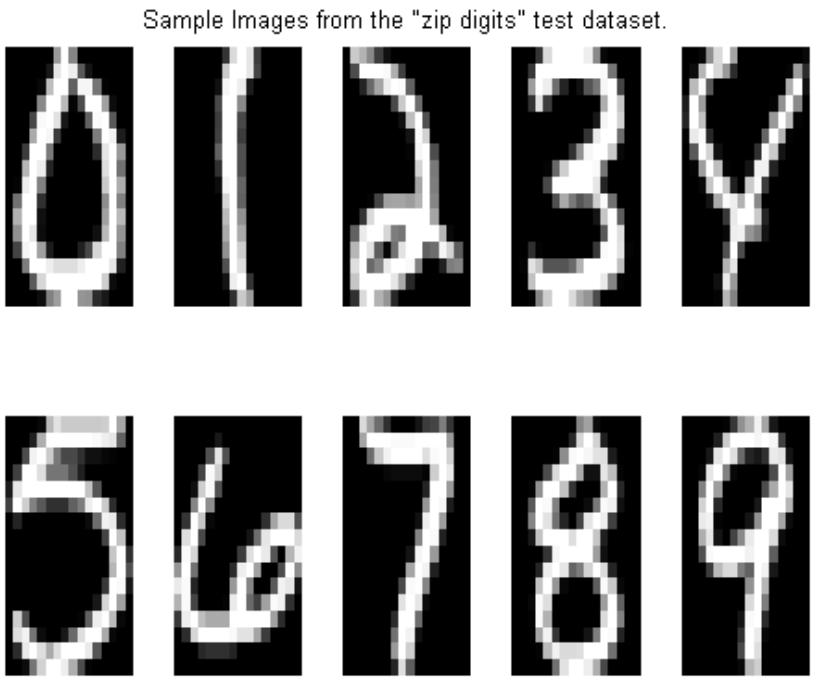


Figure 1.3. Sample images from ZIP code digits test dataset.

CHAPTER 2

IMAGE REGISTRATION FRAMEWORK

In this chapter, we derive the framework used to register two images. Since the handwritten digits tend to have smooth variations along the contours, we need to place smoothness constraints on the velocity field that registers images. Large deformation diffeomorphic metric matching is a technique which has been used to estimate a smooth velocity field that registers two images [1]. The success of this technique lies in the estimation of the deformation field with certain smoothness properties. Also, the control point parametrization gives us the feature for comparing deformations.

2.1 Registering I_{src} with I_{tar}

Let ϕ be an intensity-preserving deformation field that maps each point in the source image domain to the target image domain. Let I_{src} and I_{tar} be continuous functions in the source and target domains. Thus, the objective of registering the source image with the target is minimizing the L^2 norm between these images:

$$A(y) = ||I_{src} \circ \phi^{-1} - I_{tar}||^2 \tag{2.1}$$

$$= \sum_{k=1}^{M} (I_{src}(\phi^{-1}(y_k)) - I_{tar}(y_k))^2 \tag{2.2}$$

Let $\mathbf{c} = \{c_1, ...c_N\}$ be a finite set of control points. The deformation field is parametrized by momenta vectors at the control points $\boldsymbol{\alpha} = \{\alpha_1, ...\alpha_N\}$. The velocity field being continuous can be found at any point x in the source image domain by using a Gaussian interpolating kernel:

$$v(x) = \sum_{i=1}^{N} K(x, c_i)\alpha_i \tag{2.3}$$

$$\text{where } K(x, y) = exp(-|x - y|^2/\sigma^2) \tag{2.4}$$

The transform ϕ in this small deformation setting can be seen as $\phi(x) = x + v(x)$. It should be noted that the inverse of the field is approximated as $\phi^{-1}(y_k) = y_k - v(y_k)$. The

regularity term can be defined as the the kinetic energy of the deformation field which makes sure that the field is regularized as:

$$||v||^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i^T K(c_i, c_j)\alpha_j \tag{2.5}$$

Now we can write the objective function that we minimize in order to match the source image to the target image:

$$E(\mathbf{c}, \boldsymbol{\alpha}) = ||I_{src} \circ \phi^{-1} - I_{tar}||^2 + \gamma||v||^2 \tag{2.6}$$

$$= A(y) + \gamma||v||^2 \tag{2.7}$$

γ is the trade-off between the image fidelity term and the regularization term. The higher its value, the smoother the velocity field. We perform unconstrained line search using the gradient descent algorithm [11] on this objective to get the optimal value of momenta vectors as well as the control points. The gradient with respect to the momenta vectors can be written as:

$$\begin{aligned}\frac{1}{2}\nabla_{\alpha_i} E = &- \sum_{k=1}^{M} K(c_i, y_k)(I_{src}(y_k - v(y_k)) - I_{tar}(y_k))\nabla_{y_k - v(y_k)} I_{src} \\ &+ \gamma \sum_{j=1}^{N} K(c_i, c_j)\alpha_j\end{aligned}$$

Although we will not be using the gradient update for finding the optimal control point positions since we need to have a common basis for comparison, we mention the gradient of the objective with respect to the control point positions:

$$\begin{aligned}\frac{1}{2}\nabla_{c_i} E = &\sum_{k=1}^{M} \big(\frac{2}{\sigma^2} K(c_i, y_k)(I_{src}(y_k - v(y_k)) - I_{tar}(y_k))(\nabla_{y_k - v(y_k)} I_{src})^T \alpha_i (c_i - y_k)\big) \\ &- \gamma \sum_{i=1}^{N}\sum_{j=1}^{N} K(c_i, c_j)\alpha_i\alpha_j(c_i - c_j)\end{aligned}$$

The gradient descent uses a convergence criterion, the breaking ratio. The breaking ratio is defined as the ratio of drop in objective function value referred as the energy in the current iteration, to the drop in energy from the start of the gradient descent process. If E_0 is the value of the objective function at the beginning of gradient descent and E_n is the value at the n^{th} iteration, then the breaking ratio at the n^{th} iteration is defined in the following equation

$$Br_n = \frac{E_0 - E_n}{E_{n-1} - E_n}$$

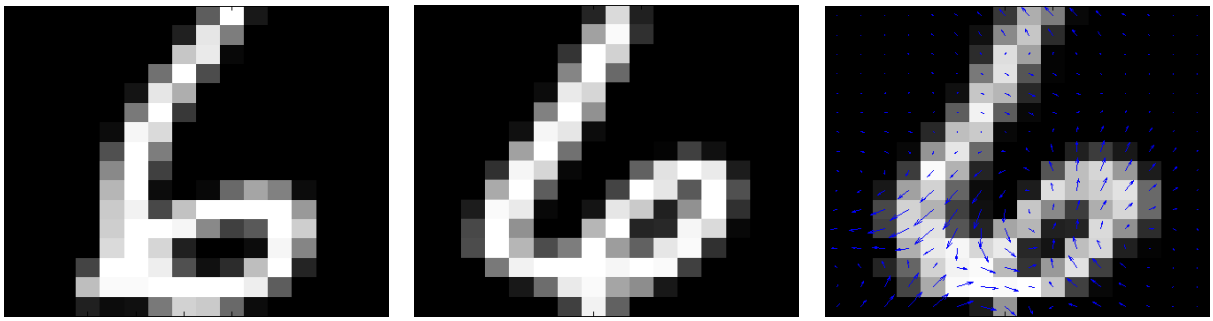
The gradient descent is terminated when the breaking ratio value is less than a predefined threshold, say Br_{th}, i.e., when $Br_n \leq Br_{th}$.

2.2 Results and conclusion

Figure 2.1 shows the result of deforming the image of a digit 6 to match another image of a handwritten 6. The two images have been taken from the training dataset. The image on the top left is the source image which we map to the target image on the top right. The registration is performed using a dense grid of 256 control points distributed regularly on the image.

The objective energy keeps decreasing and the rate of decrease is small after about 8 iterations. The total energy is the weighted sum of the L^2 difference between the deformed and target image and the regularity term governing the smoothness of the velocity field. The gradient descent stops after 21 iterations since it has satisfied the convergence criterion.

The gradient descent is stable with the given parameter settings that have been used. The control parameters that affect the result of the gradient descent are the number of control points N, the width of the Gaussian kernel σ, and the regularity trade-off γ. The control parameters that affect the rate of convergence are the breaking ratio threshold Br_{th} and the step size of the gradient descent.



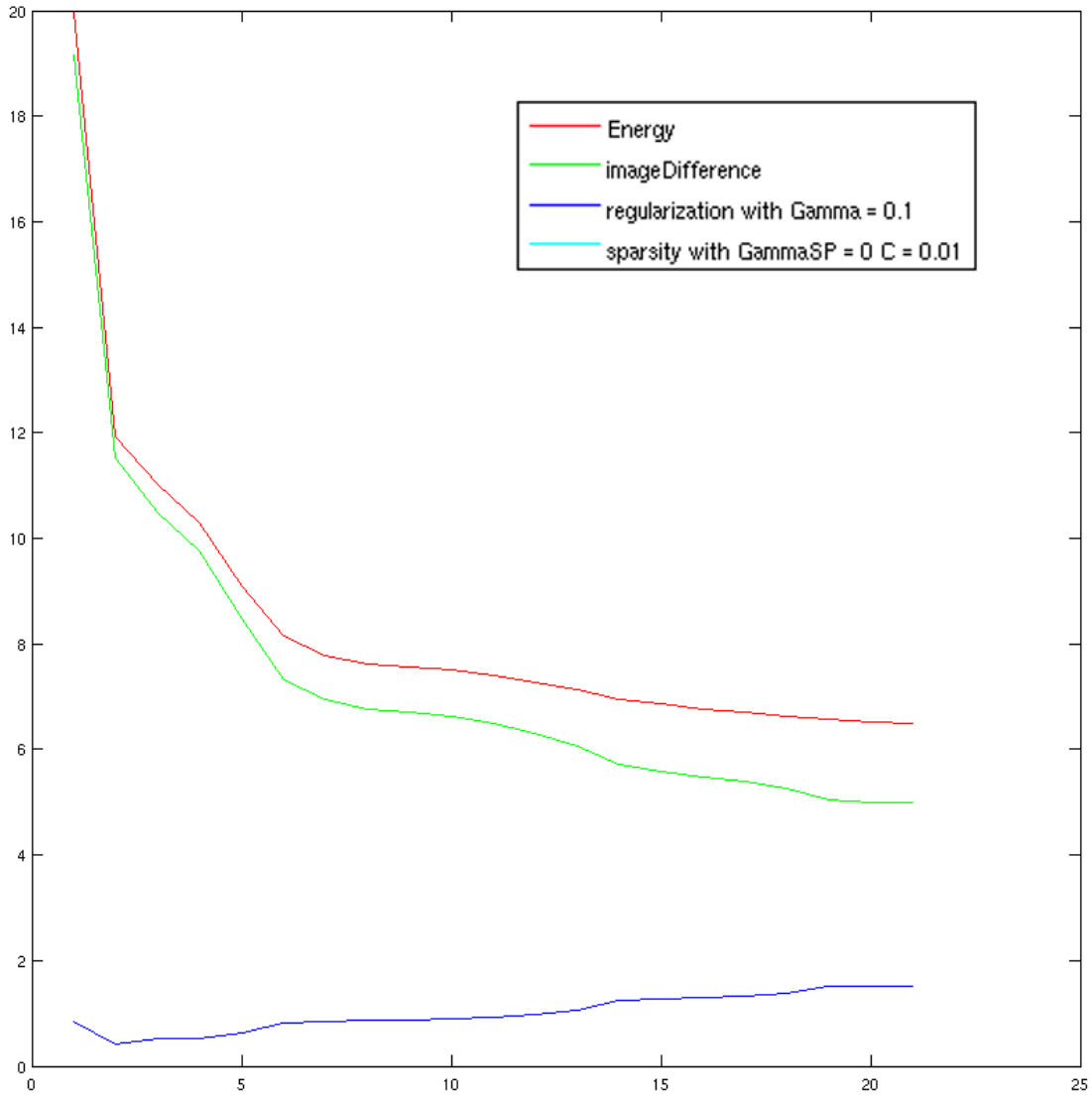


Figure 2.1. Results of deformation. **Top Left**: Source Image of digit 6. **Top Right**: Target Image of image 6. **Bottom Left**: Deformed image with 256 momenta vectors overlain over the control points. **Bottom Right**: The value of the objective against the iterations of the gradient descent.

CHAPTER 3

ATLAS ESTIMATION

In order to perform population analysis for tasks such as classification, we need to obtain a representative template of a class. This can be obtained by performing joint optimization of an objective function discussed here across the entire population of a class. We use the optimization framework discussed in [3]. The representative template image is the mean given by the L^2 norm on the space of the images of the class. This mean on this space is called a template. The atlas is a set comprising the template and a collection of deformation vectors that register the template with each of the images of the class in the dataset. In this step, we jointly optimize the template image and the deformation momenta that map each image in the class to the template to get an optimum atlas. We are also motivated to reduce the number of control points in order to reduce the size of the feature descriptor. A technique to compute an optimal set of landmarks is also discussed here.

3.1 Derivation of the atlas formation process

If I_0 is the template image of the class, $\mathbf{c}$ the set of control point vectors, and $\boldsymbol{\alpha}_i$ the set of deformation vectors that register the template with the image I_s of the class, then the objective function that we are attempting to minimize in order to find the optimal template can be written as:

$$E(I_0, \mathbf{c}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{N_s}) = \sum_{s=1}^{N_s} \{A_s(y) + \gamma ||v_s||^2\} \tag{3.1}$$

$$\text{where } A_s(y) = ||I_s \circ \phi^{-1} - I_0||^2 \tag{3.2}$$

$$v_s = \text{velocity field registering } I_s \text{ with template image } I_0 \tag{3.3}$$

The gradient of the objective with respect to the momenta is:

$$\nabla_{\boldsymbol{\alpha}_s} E = \nabla_{\boldsymbol{\alpha}_s} E_s \tag{3.4}$$

$$\text{where } N_s = \text{ total number of images of class } l \tag{3.5}$$

Also, the gradient of the objective with respect to the template image I_0 which we use in order to get a better estimate of the template I_0 is the sum of the gradient $\nabla_{I_0} A_s(y_s(0))$ over the N_s images. It can be shown to be equal to the sum of the splatted version of the residual images:

$$\nabla_{I_0} E = \delta A(\boldsymbol{y}(0)) \tag{3.6}$$

$$= \sum_{s=1}^{N_s} \Big(\text{splat } (I_0 \circ \phi_{\boldsymbol{\alpha}_s}^{-1} - I_s) \text{ into template domain} \Big) \tag{3.7}$$

We perform a gradient descent on the momenta vectors and the template image simultaneously in order to get the optimal template image and optimal deformation momenta vectors. This is a straightforward method of performing gradient descent on the given objective to find the template image. Note that we have not mentioned the update to the control point position since we will not use it hereafter.

3.1.1 Atlas formation using iterative averaging

Another method to update the atlas is using iterative averaging. Here we take the objective as defined in 3.1 but do not compute the gradient with respect to the template image. The objective is not a function of the template image and hence is defined as in 3.8.

$$E(\mathbf{c}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{N_s}) = \sum_{s=1}^{N_s} \{A_s(y) + \gamma ||v_s||^2\} \tag{3.8}$$

$$\text{where } A_s(y) = ||I_0 \circ \phi^{-1} - I_s||^2 \tag{3.9}$$

$$v_s = \text{velocity field registering } I_s \text{ with template image } I_0 \tag{3.10}$$

In order to compute the template image, we start with an initial estimate of the template image I_0 at iteration 0. Next, we register all the images in the class with this estimate of the class template. The average of the deformed images is the new estimate of the template as given in 3.11.

$$I_0 = \frac{1}{N_s} \sum_{s=1}^{N_s} I_s \circ \phi_{\boldsymbol{\alpha}_s} \tag{3.11}$$

This process is repeated till the objective defined in 3.8 is less than the predefined breaking ratio.

3.1.2 Results

We have run the atlas formation procedure using both splatting as well as averaging. Figure 3.1 shows the results of running the atlas formation using both the techniques for two

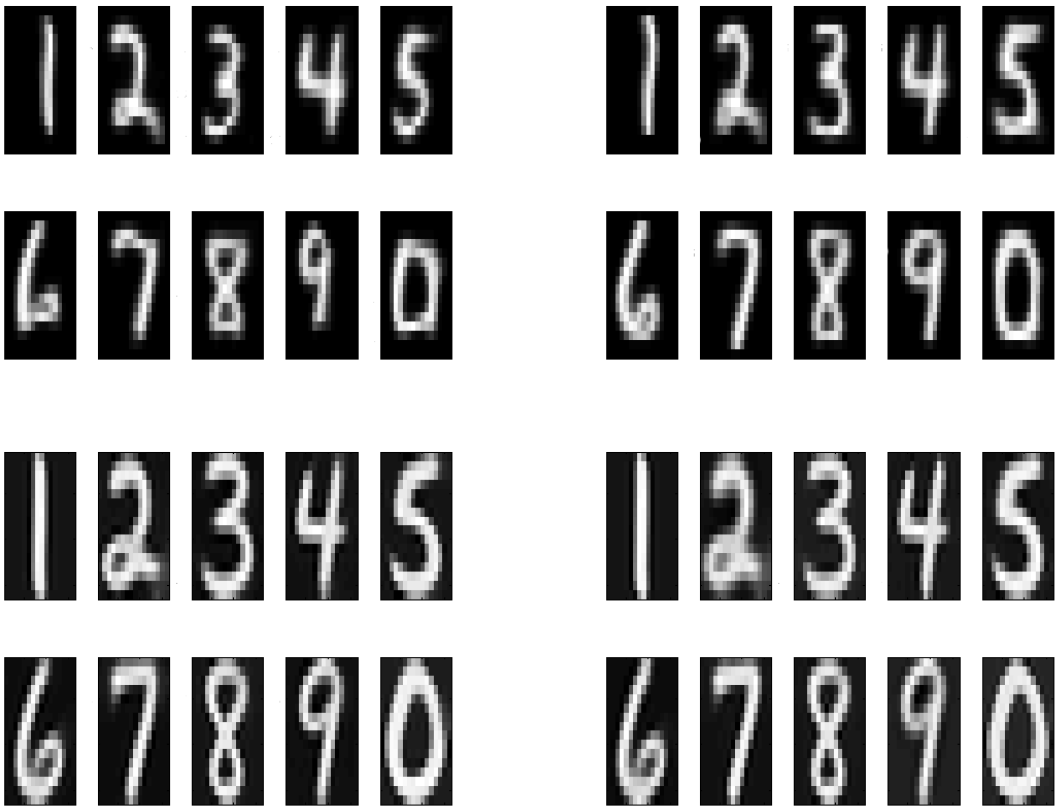


Figure 3.1. Results of atlas formation. **Top Left:** Template images in atlas formed using averaging for $\sigma = 1$. **Top Right:** Template images in atlas formed using averaging for $\sigma = 3$. **Bottom Left:** Template images in atlas formed using splatting for $\sigma = 1$. **Bottom Right:** Template images in atlas formed using splatting for $\sigma = 3$.

values of the kernel width σ. As can be seen from Figure 3.1, the averaging technique tends to shrink the template compared to the actual images in the dataset itself. This behavior is possibly due to the regularization term reducing the magnitude of the momenta vectors, leading to an update in the template that is smaller than it should be. Due to the shrunk template images, the result of the averaging technique was not appealing since they are not characteristic means of the images.

The result of splatting is better and has been prescribed in [3] as well. The only downside of splatting is that the template images have negative values. This is due to large gradient steps which tend to decrease the overall objective but result in negative values for some pixels. This behavior was remedied by adaptive change in the step size which is a form of gradient descent line search described in [11].

For further discussion, let us denote Θ_l as the set of all the momenta vectors that deform the mean image of the class l to the images of class l in the dataset. Let I_{li} denote the image i of class l in the dataset, μ_l denote the mean image of class l, $\phi_{\boldsymbol{\alpha}}$ be the deformation field parametrized by the momenta vectors $\boldsymbol{\alpha}$. Thus, we can define:

$$\Theta_l = \{\boldsymbol{\alpha}_i | \mu_l(\phi_{\boldsymbol{\alpha}_i}(x)) \approx I_{li}(x)\} \tag{3.12}$$

Note that the deformed mean image $\mu_l \circ \phi_{\boldsymbol{\alpha}_i}$ is not exactly equal to the target image I_{li} since the registration process does not exactly match the two.

3.2 Computing an optimal set of landmarks

We will be classifying the images based upon the deformation that is required to match the template image of each class with the test image. We can compare deformation fields using the momenta vectors $\boldsymbol{\alpha}$ that parametrize each of the deformation fields. In order to compare the momenta vectors, they need to be defined at the same set of control points. Thus, we cannot move the control points in any step of the entire process.

We can always place the control points on a regularly spaced grid. However, since we cannot move the control points, there are two issues we face. The first issue is that we need to have a reasonably dense distribution of control points in order to capture the variations in the data. If we increase the number of control points in order to increase this density, then the number of feature vectors goes up. Next, we need to capture any deformation possible from any template source image to any image in the database. Capturing a deformation implies that in a given region in the image domain which would require a deformation to match some source image in the dataset or atlas to some other image in the dataset, we would need a control point in that region.

Since we are interested only in deformations within images of a single class and not outside, we need to find out all the possible deformations that can occur between the template (which we can approximate with the mean image) and all the images of a class in the dataset. To find this, we place control points at all the grid locations in the image and register the mean image of a class with each image of the class in the dataset. The variance of the momenta vectors will tell us which control points tend to have the most varying momenta. Such points are valid candidates for being control points. We find such high variance points for each class in the entire dataset and take a union of all such sets to get the final set of control points. The process to do the above is as follows.

Let us denote Θ_l as the set of all the momenta vectors that deform the mean image of the class l to the images of class l in the dataset, assuming that we have a control points at each grid element or pixel in the image. Let I_l^i denote the image i of class l in the dataset, μ_l denote the mean image of class l and $\phi_{\boldsymbol{\alpha}}$ the deformation field parametrized by the momenta vectors $\boldsymbol{\alpha}$. Thus, we can define the set Θ_l as in 3.13.

$$\Theta_l = \{\boldsymbol{\alpha}_i | \mu_l(\phi_{\boldsymbol{\alpha}_i}(x)) \approx I_l^i(x)\} \tag{3.13}$$

The deformed mean image $\mu_l \circ \phi_{\boldsymbol{\alpha}_i}$ is not exactly equal to the target image I_{li} since the registration process does not exactly match the two.

The L_2 norm of the variance of the momenta vectors defined for each grid point(pixel) over the set Θ_l can be obtained as:

$$||\Sigma_l^2||(x) = ||\mathbb{E}[(\boldsymbol{\alpha}_i - \mathbb{E}[\boldsymbol{\alpha}_i])^2]||(x) \tag{3.14}$$

$$\text{where } \mathbb{E} \text{ acts on all the momenta vectors } i \text{ for class } l \tag{3.15}$$

This value is defined for each pixel position x over the image domain for each class l. The images in Figure 3.2 show the L_2 norm of variance for different values of the kernel width σ used for the interpolation kernel $K(x, y)$ defined in 2.4.

As can be seen from the first image in Figure 3.2, the smaller kernel tends to give us a better judgment of which pixels have high variance. Intuitively, it can be seen that the variance should be on the boundary of the main contour of any handwritten digit which is what we see for smaller values of σ.

To find the optimum position of the control points, we perform a form of discrete peak detection that tells us which pixels have the largest variation of deformation vectors for each class. In this process, for each grid location, we check to see if it has a value greater than all its neighbors. If so, it is a valid peak. Figure 3.3 shows the result of the peak detection operation.

Figure 3.2. Variance of norm over atlas. **Top to bottom, Left to right:** L_2 norm of variance of $\phi_{\boldsymbol{\alpha}_i}$ for the interpolating kernel width $\sigma = 0.5, 1, 2, 3$, and 4. The variance tends to be more distributed over the entire image domain as we increase the value of σ.

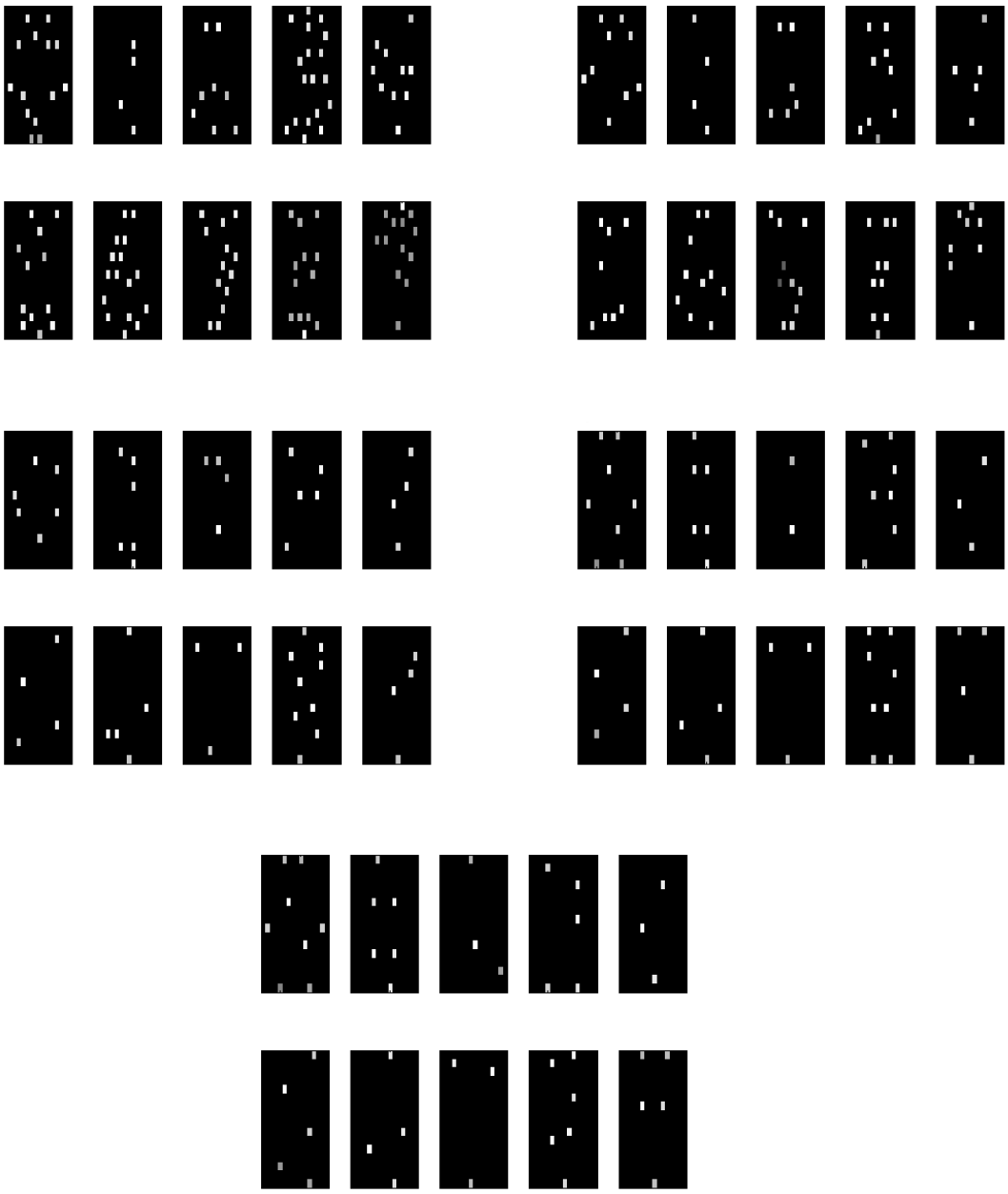


Figure 3.3. Peaks of variance norm. **Top to bottom, Left to right:** Variance peaks for the interpolating kernel width $\sigma = 0.5, 1, 2, 3$, and 4. The peaks seem to hug the contours from the outside.

As can be seen in Figure 3.3, the number of potential candidates for being control points decreases as the kernel width increases. For $\sigma = 0.5$, we have the highest number of control points.

We have repeated the procedure of peak finding on the sum of the L_2 norm of variance images for each class to get the final set of control points which is a union of the peaks found in the earlier step. The results of performing this step are shown in Figure 3.4.

As can be seen, the peaks found with lower values of σ tend to be more well distributed and the number is larger.

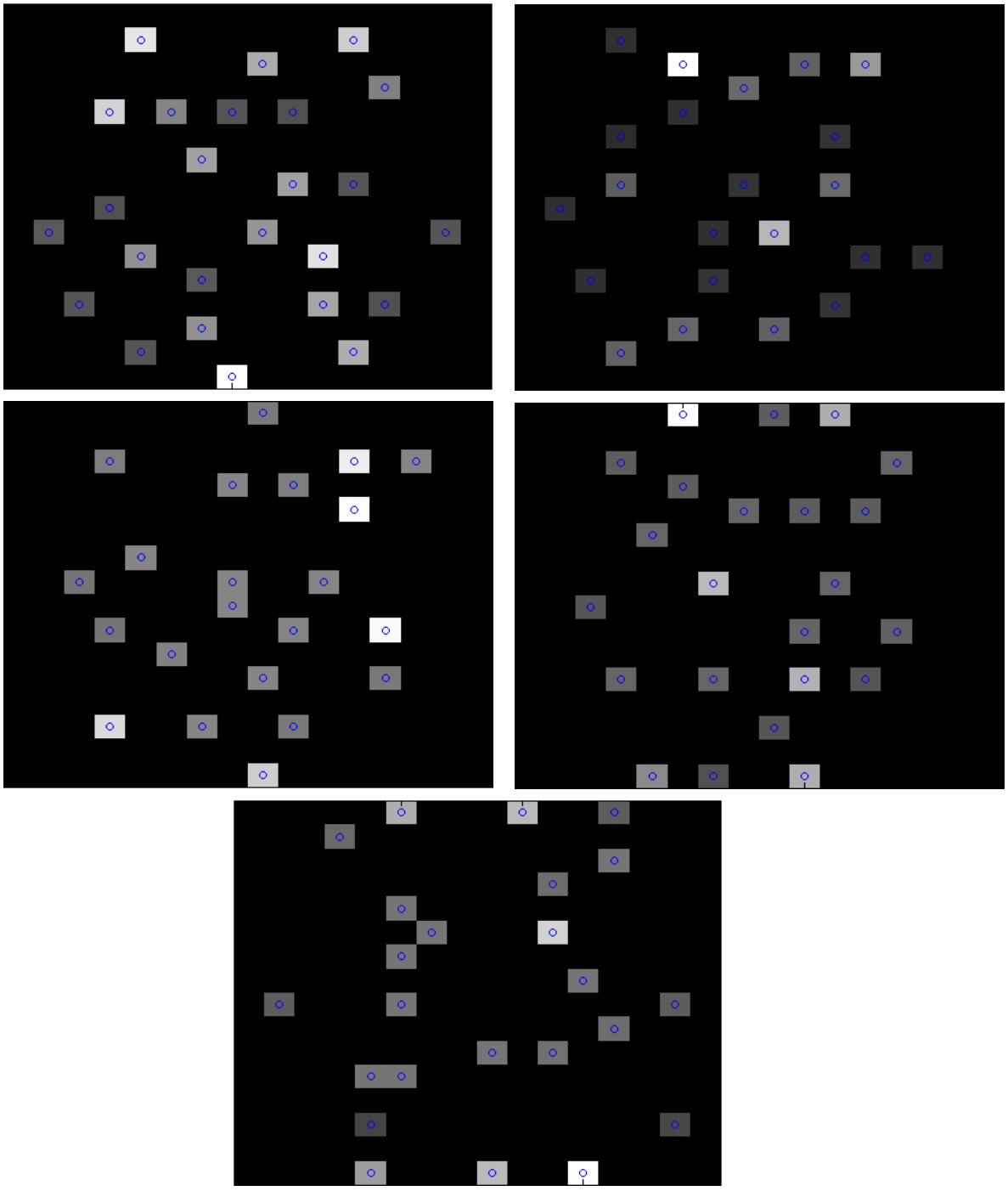


Figure 3.4. Union of variance peaks across classes. **Top to bottom, Left to right:** Peaks found as a union of the peaks from the previous steps.

CHAPTER 4

BINARY CLASSIFICATION

To verify the performance of momenta as the feature vector, we will construct a binary classifier. It should be noted that in order to compare velocity fields mapping the template images of different classes to the test image, the control points need to be at the same location. In this chapter, we will discuss the binary classifier to distinguish between two classes. We will discuss the various classification criteria, the receiver operating characteristics used to compare binary classifier performance, the effect of varying the classifier and feature parameters, and the effect of changing the size of the training dataset.

4.1 Classification criteria

We have mainly experimented with three classification criteria. Each criterion uses a metric that defines a distance from the decision boundary. Following is a description of the metrics and the criteria they imply:

1. **Mahalanobis distance**: We assume that the training data for class l is the deformation field that registers the template of class l with each of the subject images in the training dataset of class l. Thus, the training data for class l consist of a set $\Theta_l = \{\boldsymbol{\alpha}_i\}_l$ as described in equation 3.13. Now, to classify the test image, we register the template of the class l with the test image using the technique described in Chapter 2, giving us the deformation field $\boldsymbol{\alpha}^l_{test}$. Let S denote the covariance matrix of the set of deformation momenta vectors of class l in the set Θ_l which is defined in equation 4.1

$$S = \mathbb{E}[(A - \mathbb{E}(A))(A - \mathbb{E}(A))^T] \tag{4.1}$$

$$\text{where } A = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_n \end{bmatrix} \tag{4.2}$$

Then the Mahalanobis distance of the test deformation μ_l from the mean deformation Θ_l can be given by:

$$M_l(\boldsymbol{\alpha}^l_{test}) = \sqrt{(\boldsymbol{\alpha}^l_{test} - \mu_l)S_l^{-1}(\boldsymbol{\alpha}^l_{test} - \mu_l)} \tag{4.3}$$

The Mahalanobis distance tells us how many standard deviations our test deformation is from the mean deformation of the class. The closer to the mean deformation of a class the test deformation is, the more likely is it to belong to that class. As can be seen, the Mahalanobis distance is a normalized metric. It is normalized by the variance of the training set A of each class as seen in 4.1. Thus, the classification criterion we have used is the smallest Mahalanobis distance, which can be written as given in equation 4.4.

$$\hat{y} = \arg\min_{l\in\{1,2\}} M_l(\boldsymbol{\alpha}^l_{test}) \tag{4.4}$$

The Mahalanobis distance is used to account for the correlations within a dataset as described in [10]. It is invariant to scale and is referred to as a normalized Euclidean distance.

2. **Magnitude of momentum**: If a test image belongs to a class, the deformation that maps the template image of that class to it should be small. This implies that the L^2 norm of the momenta characterizing it should be small, compared to the deformation required to map the template image of any other classes to it. Thus, assuming the notation discussed in the criterion 1 above, the classification criterion can be written as in equation 4.5.

$$\hat{y} = \arg\min_{l\in\{1,2\}} ||\boldsymbol{\alpha}^l_{test}||_{mag} \tag{4.5}$$

$$\text{where } ||\boldsymbol{\alpha}^l_{test}||_{mag} = \sqrt{\sum_i ||\alpha_i||_{L^2}} \tag{4.6}$$

This is written assuming the binary classifier between the classes l we are comparing against.

3. **Using the data matching term**: The data matching criterion uses the accuracy of the registration as a means to classification. If a template of a class registers accurately with the test image, then the test image would belong to the specific class. Let us say that the deformation obtained by registering the template of class l with the test image is a function of the momenta $\phi^{-1}(\boldsymbol{\alpha}^l_{test})$. Then, applying the deformation to the template gives us an image which is closely matched to the

test image. Based on how well the two images match, we propose the classification criterion in equation 4.7:

$$\hat{y} = \arg\min_{l\in\{1,2\}} ||I^l_{template} \circ \phi(\boldsymbol{\alpha}^l_{test}) - I_{test}||_{L^2} \tag{4.7}$$

This is based upon the quality of the registration. If the deformation found results in the template image closely matching the test image, the test image belongs to the class. In order to measure how closely the test image matches the deformed template image, we take the L^2 norm of the difference between the two as seen in equation 4.7. This is the criterion which has yielded the best results, as can be seen in following sections.

4.2 Receiver operating characteristics plots

The performance of binary classifiers can be visualized and measured using Receiver Operating Characteristic plots, also referred to as ROC curves from here onwards. They can be used for the selection of internal parameters in classifiers. In our case, we will vary the parameters σ, γ, the number of control points, and the number of training samples to find out their effect on the classifier. The ROC curve is plotted by changing the classification threshold between the classification distances in binary classifiers. The traditional binary classifier can be written as given in 4.8.

$$\hat{y} = 1 \text{ if } d(\boldsymbol{\alpha}^1_{test}) \leq d(\boldsymbol{\alpha}^2_{test}) \tag{4.8}$$

$$= 2 \text{ if } d(\boldsymbol{\alpha}^1_{test}) > d(\boldsymbol{\alpha}^2_{test}) \tag{4.9}$$

Here, the function d is any distance metric which has been discussed in section 4.1. In order to plot the performance of the classifier, we introduce a threshold term δ which changes the classifier equation to 4.10:

$$\hat{y} = 1 \text{ if } d(\boldsymbol{\alpha}^1_{test}) - \delta \leq d(\boldsymbol{\alpha}^2_{test}) \tag{4.10}$$

$$= 2 \text{ if } d(\boldsymbol{\alpha}^1_{test}) - \delta > d(\boldsymbol{\alpha}^2_{test}) \tag{4.11}$$

As can be seen in the above equation, varying the threshold δ will result in a shift in the linear classifier boundary. When testing the classifier, if we plot the results of the true positive rate (TPR) against the false positive rate (FPR) for various values of δ, we get the ROC curve which quantifies the quality of the classifier. Finally, the classification criterion can be given by 4.12

$$\hat{y} = 1 \text{ if } d(\boldsymbol{\alpha}^1_{test}) - d(\boldsymbol{\alpha}^2_{test}) - \delta \leq 0 \tag{4.12}$$

$$= 2 \text{ if } d(\boldsymbol{\alpha}^1_{test}) - d(\boldsymbol{\alpha}^2_{test}) - \delta > 0 \tag{4.13}$$

The ROC graphs are two-dimensional graphs in which TP rate is plotted on the Y axis and FP rate is plotted on the X axis. Each point in the ROC graph represents the performance of a single classifier. If we vary δ in the above set of equations, then we get a ROC curve that tells us the performance of the classifier for a given set of parameters. The graph denotes the relative trade-off between true positives and false negatives. The closer the graph to the top left corner and the larger the area under the ROC curve, the better the performance of the classifier. Some of the properties of the ROC graphs that are attractive are its insensitivity to class skew, and ease of comparison of classifiers by the metrics area under the curve as well as ROC average comparison [5]. We use the algorithm for computing the area under the curve and the average of the curve that has been discussed in [5].

The binary classifier using metrics discussed in section 4.1 has been implemented for classification between two sets of digits. The first binary classifier is between the digits 1 and 3 while the second set discusses results of the binary classifier between digits 2 and 5. The major motive of this experiment is to decide the optimal value of σ and γ which needs to be used for classification. Also, the effect of varying the number of training samples and the dimensionality of the deformation descriptor which is the number of control points has been discussed. We will use ROC plots to measure and compare the performance of classifiers. Note that all of the following tests use a regular grid distribution of control points and the Mahalanobis distance metric from equation 4.4 for classification.

4.3 Effect of varying σ and γ_r

Let us plot the ROC curves for the binary classifier between digits 1 and 3 for $\sigma = \{1, 2, 3, 4, 5\}$ and $\gamma_r = \{0.001, 0.01, 0.1, 0.5, 0.9\}$. Figure 4.1 and Figure 4.2 show the ROC plots for the classifier between digits 1 and 3 varying σ and γ.

The plot makes it difficult to compare the various curves and find a good operating point. Instead, we can plot the different curves one for each value of γ_r, for each value of σ in order to find which value of σ is ideal. This can be done by finding the value of σ for which the 2D ROC curves have a low variance. Although a variance metric can be used to do the same, a visual inspection of the ROC curves gives us a good idea of the correct value of σ that can be selected. Similarly, we can find a good value of γ_r. The curves which helped us conclude the ideal value of $\sigma = 2$ are shown in Figure 4.3. Although the curves for the other values of σ are not shown here, it can easily be seen that the variance of the 5 curves is small. It is smaller than the variance of the curves for other values of σ. This

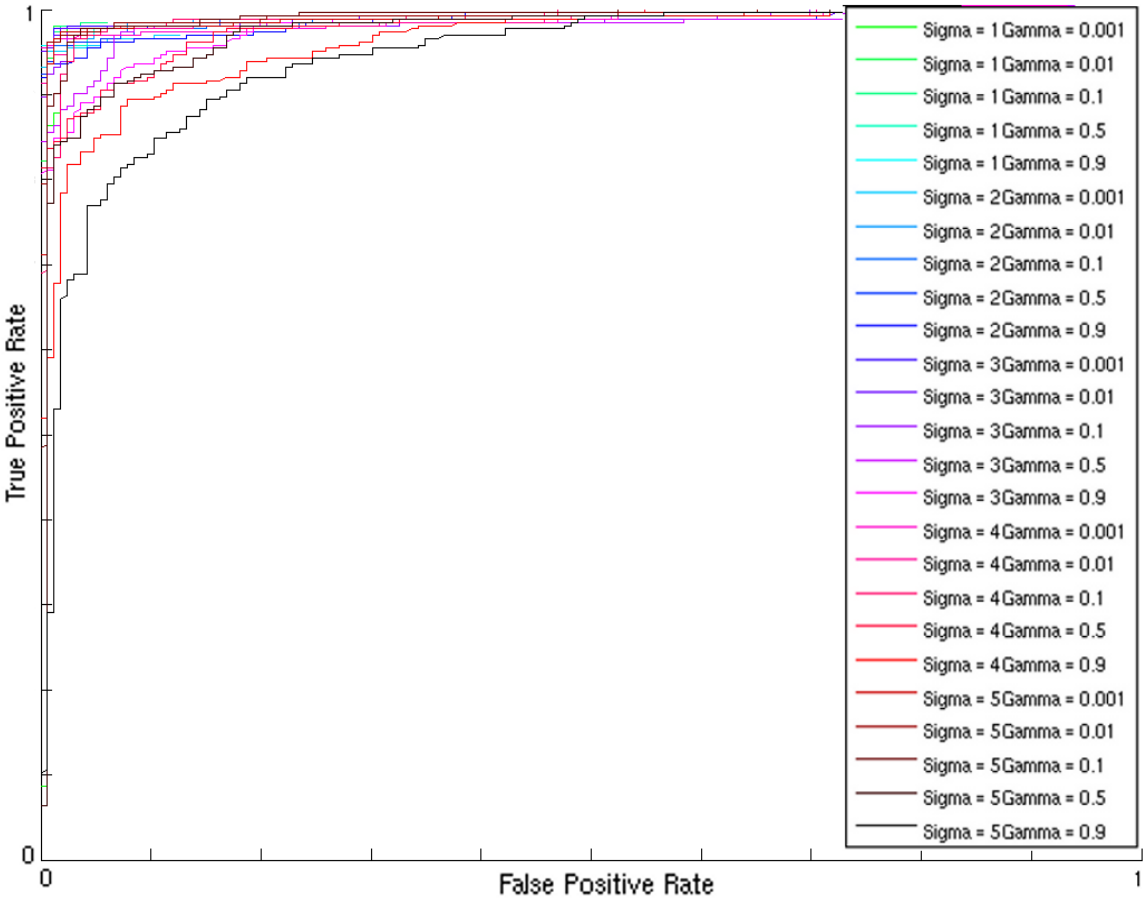


Figure 4.1. ROC curve for classification between digits 1 and 3.

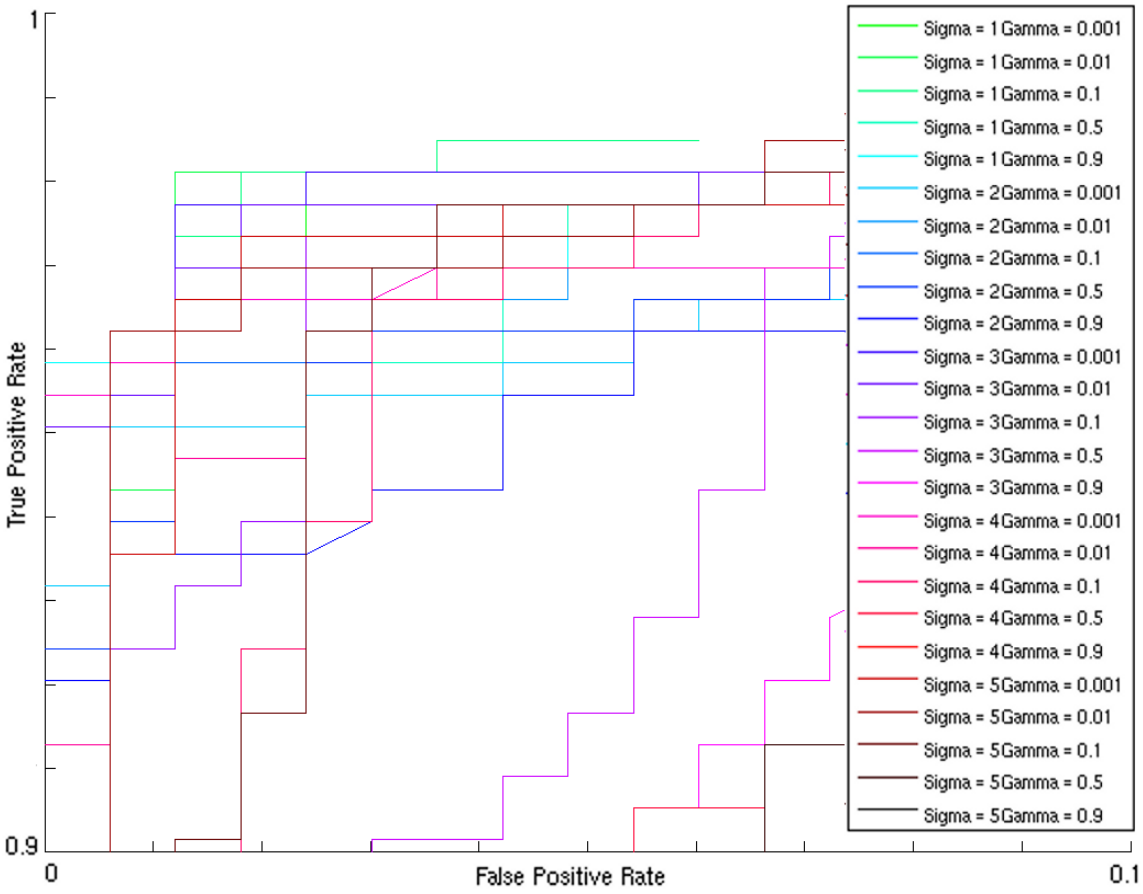


Figure 4.2. Magnified version of the plot, magnified for FPR between 0 and 0.1 and TPR between 0.9 and 1.

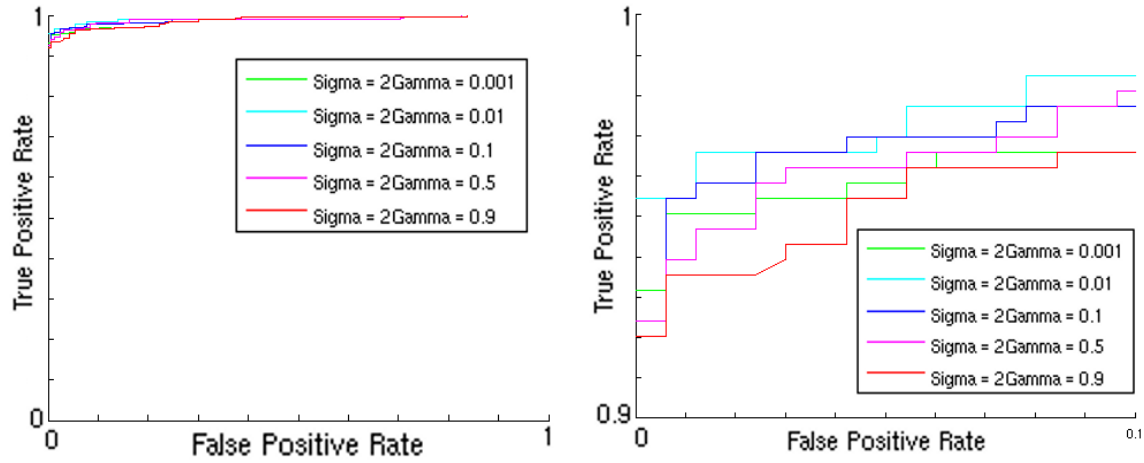


Figure 4.3. Roc curves of classification between digits 1 and 3 for $\sigma = 2$. **Left:** ROC curve. **Right:** Magnified version of the same plot, magnified for FPR between 0 and 0.1 and TPR between 0.9 and 1.

led to the conclusion that $\sigma = 2$ is a good operating point for the binary classifier.

We will use the same technique of visual inspection to find the value of γ_r. Plotting the ROC curves for γ_r for different values of σ and comparing the variance tells us that $\gamma_r = 0.01$ has the lowest variance. Figure 4.4 shows the ROC curves used to come to this conclusion.

Thus, using Figures 4.3 and 4.4, we can conclude that the binary classifier between digits 1 and 3 has an ideal operating point at $\sigma = 2$ and $\gamma_r = 0.01$.

For the digits database, we wanted to test the hypothesis that the same kernel width and regularity penalty does not give similar results for binary classifiers between other classes of images. This was done by building a binary classifier between the digits 2 and 5 and repeating the above tests. Figure 4.5 and 4.6 show the various ROC curves plotted for various values of σ and γ_r for the binary classifier between digits 2 and 5.

A preliminary inspection of the results tell us that the binary classifier between digits 2 and 5 does not perform as well as the classifier between digits 1 and 3. Similarly, plotting the values of σ and γ_r, we concluded that $\sigma = 3$ and $\gamma_r = 0.1$ is a good operating point for this binary classifier. Hence, the same value of kernel width and regularity penalty does not give good classification results.

In order to verify our results for the binary classifier between digits 1 and 3, we find the area under the curve. The larger the area under the curve, the higher the accuracy. Figure 4.7 shows the plot for all the values of σ and γ_r along with the area under the curve. As we can see, the area under the curve for $\sigma = 2, \gamma_r = 0.01$ has $AUC = 0.98729$ which is high and hence reinforces our results.

We can conclude that the value of the optimal σ corresponds to the mean width of the curves of the handwritten digits. Hence, the optimal σ for the classifier between digits 1 and 3 has a lower σ value of 2 while the classifier between digits 2 and 5 has an optimal value of 3. This is due to the fact that the images of the digit 2 in the dataset consistently have a small loop on the lower left which causes the average width of the digit to increase.

4.4 Effect of varying the dimensionality of the deformation descriptor

To test the effect of dimensionality of the shape descriptor on classification, we will plot the ROC curves with the same setup as used before but by changing the number of control points used. We will form the atlas of the two digits using $4 \times 4 = 16$, $5 \times 5 = 25$ up to $8 \times 8 = 64$ control points and verify the classification error using the ROC graphs. Figure 4.8 shows the desired plot. The above plot tells us that there is an optimal density

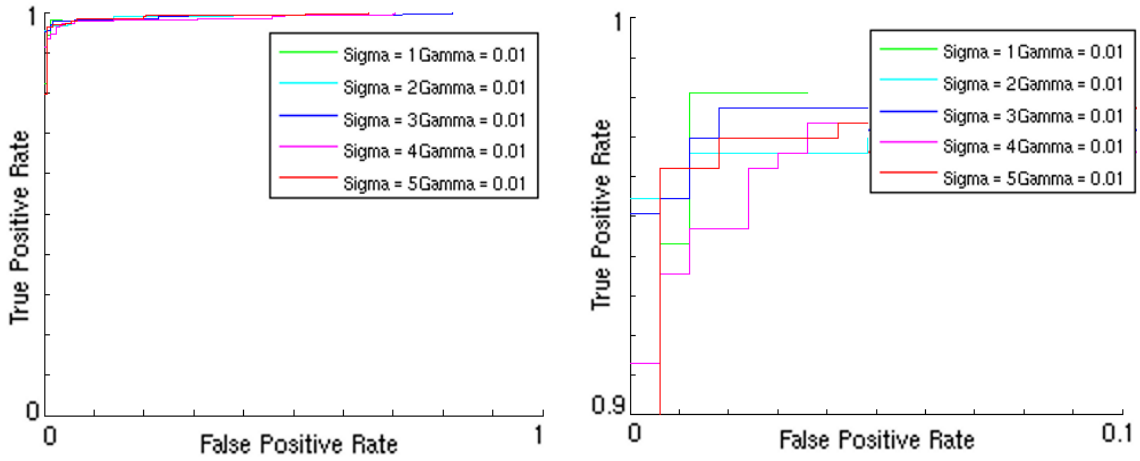


Figure 4.4. ROC curve for classification between digits 1 and 3 for $\gamma_r = 0.01$. **Left:** ROC curve. **Right:** Magnified version of the same plot, magnified for FPR between 0 and 0.1 and TPR between 0.9 and 1.

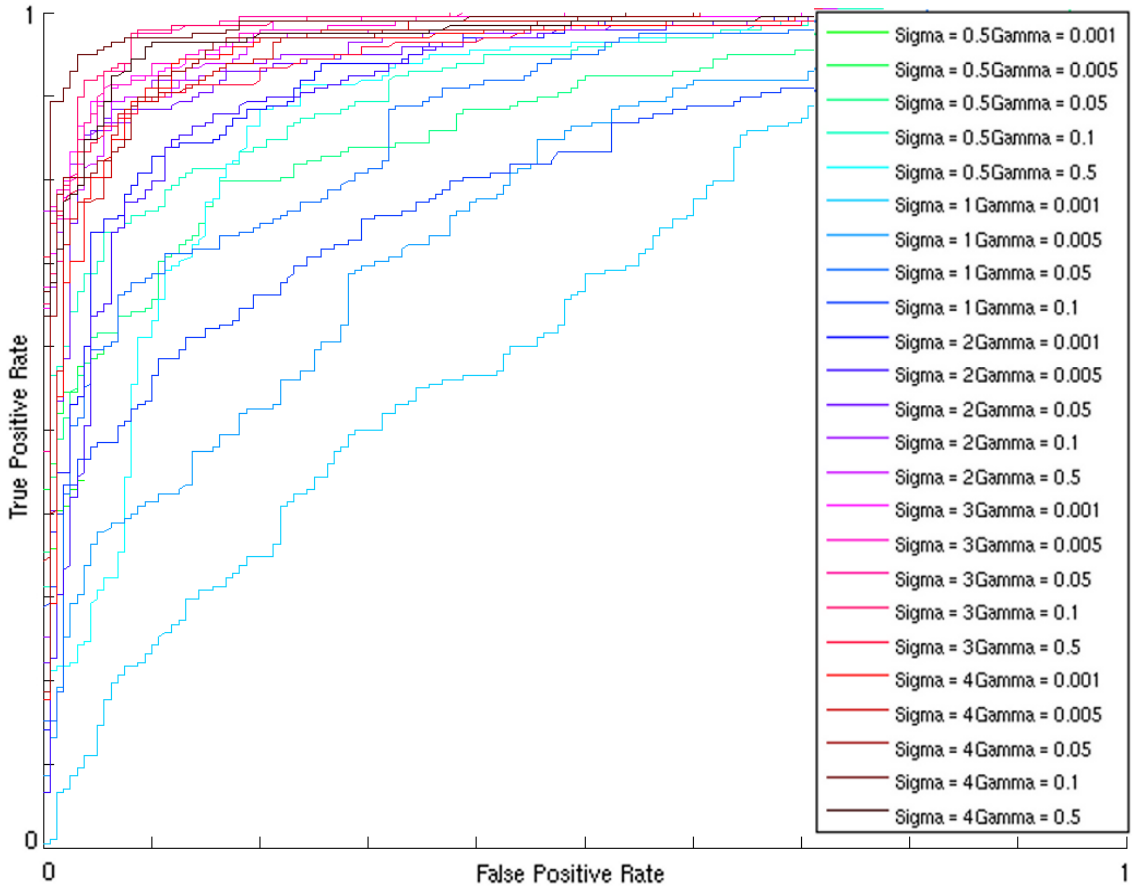


Figure 4.5. ROC curve for classification between digits 2 and 5.

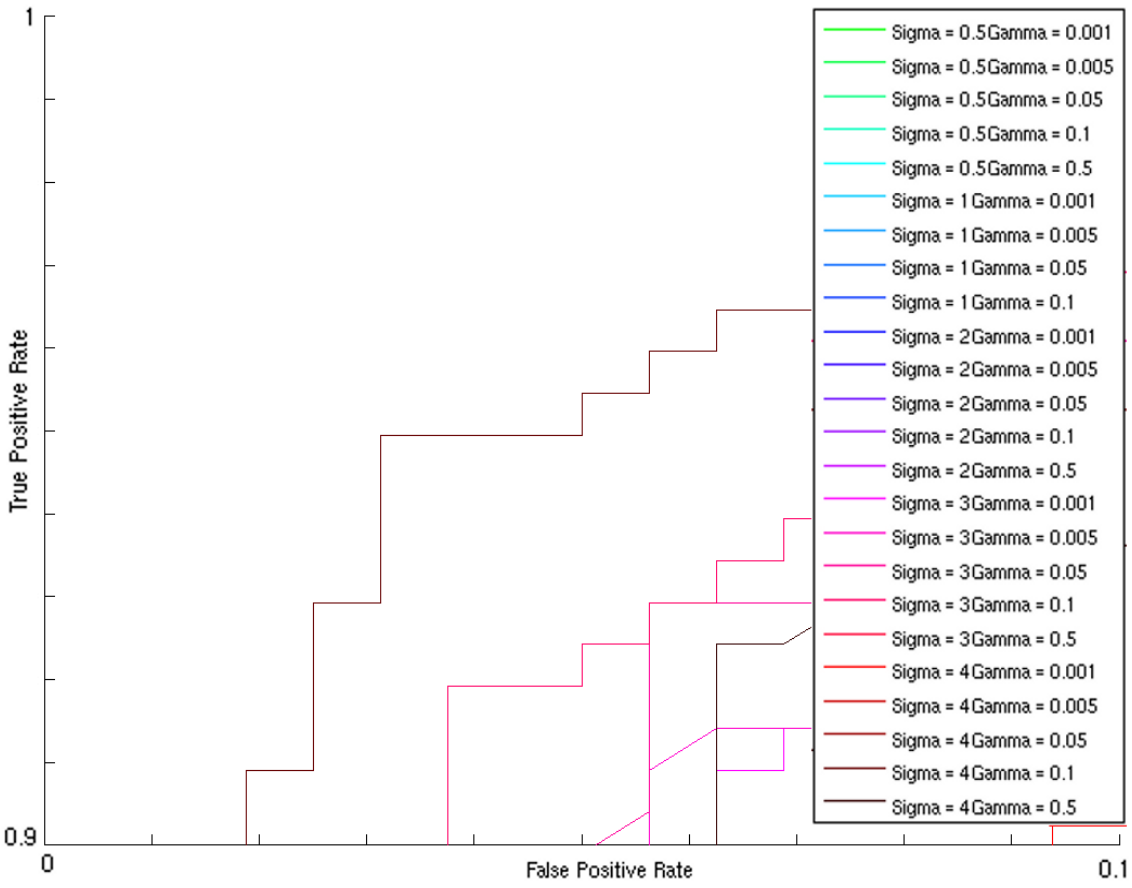


Figure 4.6. Magnified version of the same plot, magnified for FPR between 0 and 0.1 and TPR between 0.9 and 1.

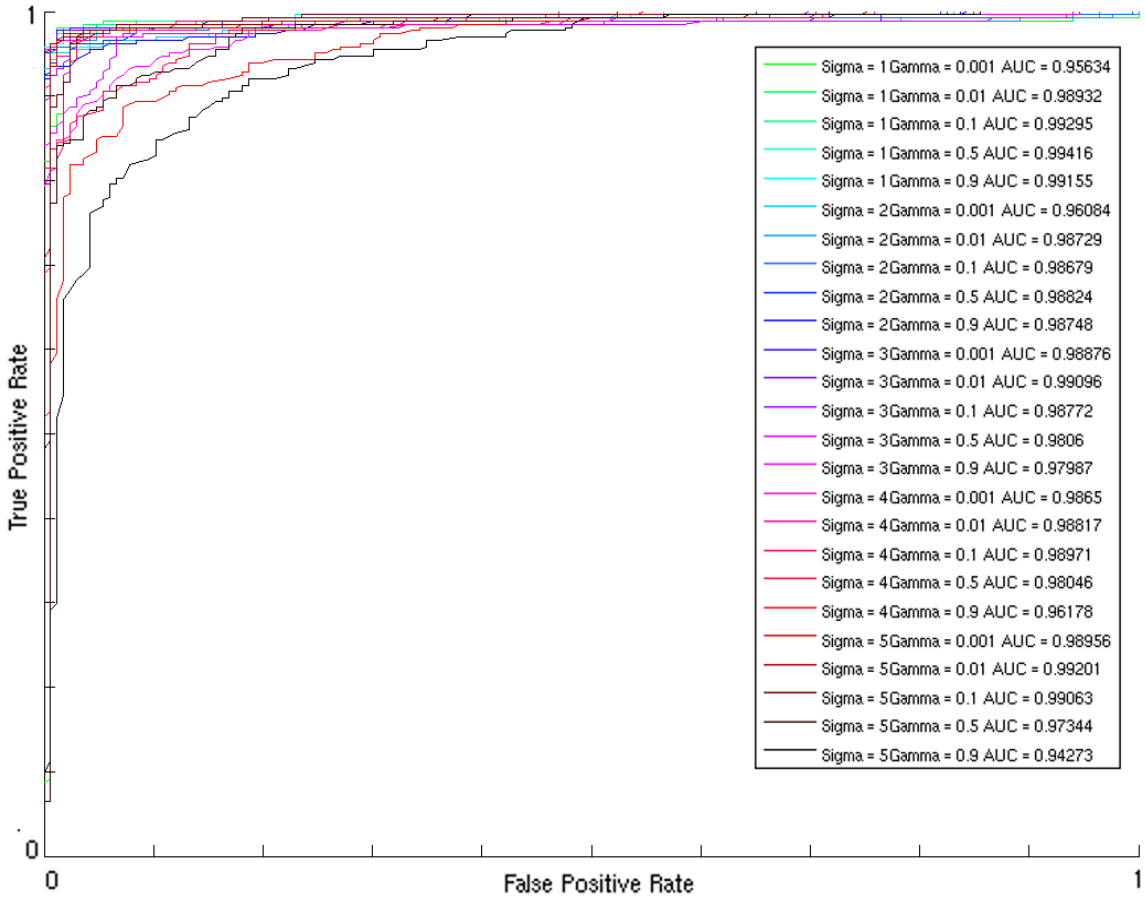


Figure 4.7. ROC curve for classification between digits 1 and 3 along with the area under the curve denoted by AUC.

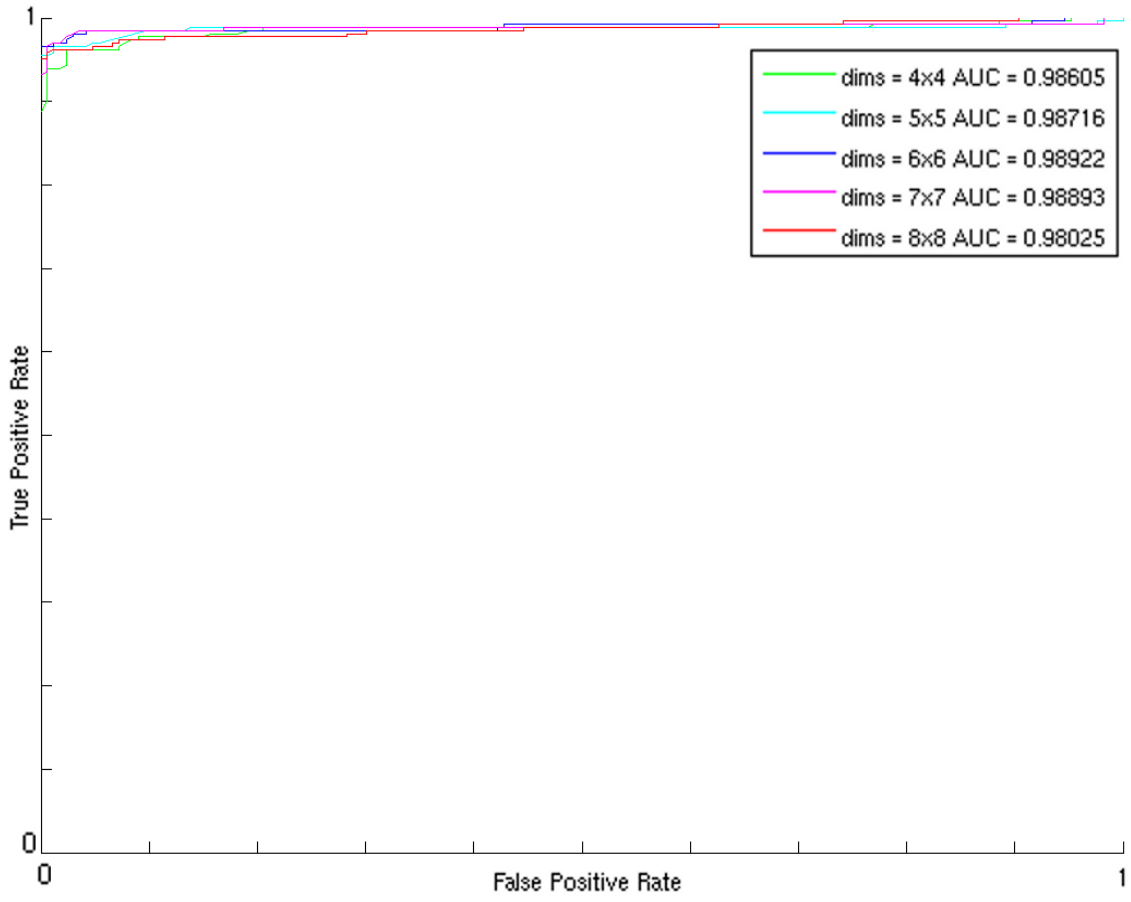


Figure 4.8. ROC curves for the binary classifier between digits 1 and 3 for different dimensionality of the classifier.

of control points that give us good classification results. If we increase or decrease this density of control points, the error rate increases. This density of control points is directly related to the kernel width σ and regularity γ_r. This can be inferred from the fact that we have used a regular distribution of points on a 5×5 grid of 25 control points to find the optimum value of $\sigma = 2$ and $\gamma_r = 0.01$ and that the optimal number of control points is 25, as can be seen in Figure 4.9. The graph gives a trend of the number of control points against the Area under the ROC curve. Figure 4.9 tells us that the area under the ROC curve is highest for the 6×6 regular grid of 36 control points. To confirm the relation of the density of control points with σ and γ_r, we will have to find the optimum value of σ and γ_r for a different density of control points which we will not attempt to do here.

In order to see if the results are consistent, let us have a look at the ROC curve as plotted for the binary classifier between digits 2 and 5. We have a slightly different trend for the area under the ROC curve as seen in the Bottom plot of Figure 4.10, but there is still an optimal value for the number of control points required, which is 25 according to this graph. The data-point 36 control points is an outlier in this case.

This leads us to the conclusion that the classifier accuracy is high for an optimal number of control points. If we increase or decrease the size of the feature vector which is the deformation descriptor, the classifier accuracy decreases. Hence, in order to obtain good classifier accuracy, we need to optimize the control point density, i.e., their number and their placement in the domain.

4.5 Effect of varying the number of training examples used in atlas formation

We want to find out the effect of sparsity of the deformation descriptor on the classification rate as the number of training samples change. To test this out, let us have a look at the effect of changing the number of training examples for a fixed number of control points. As before, we will test on a regular grid of $5 \times 5 = 25$ control points. Let us change the number of training examples in this setting for $\sigma = 2$ and $\gamma_r = 0.01$ and observe its effect on the ROC curves as shown in Figure 4.11.

From Figure 4.11, the effect of the number of training examples on classifier accuracy is not exactly clear. To clarify this, let us plot the area under each ROC curve against the number of training examples. Figure 4.12 shows the same.

Figure 4.12 tells us that as the training examples increase, the classifier gives better results. No clear relationship can be identified other than the fact that the classification

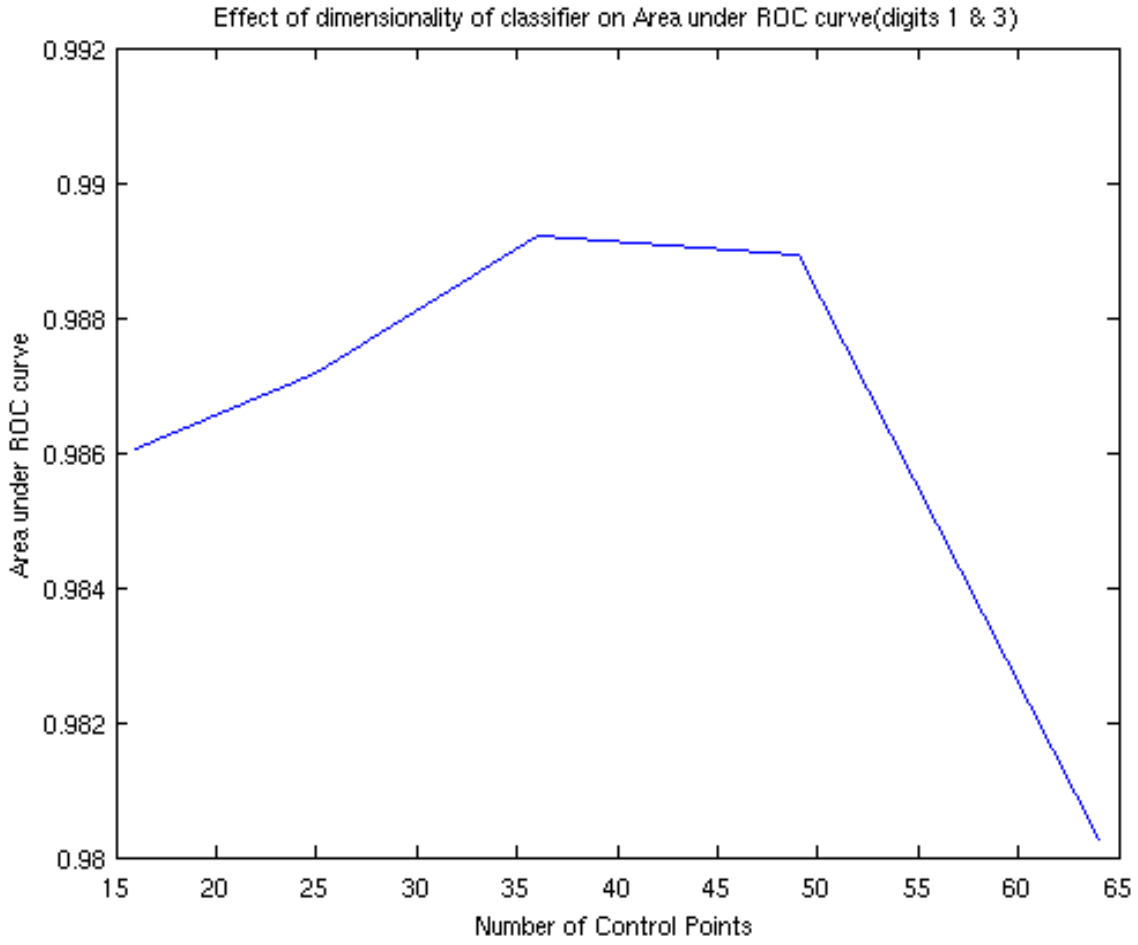


Figure 4.9. Effect of changing the number of control points on the Area under the ROC curve.

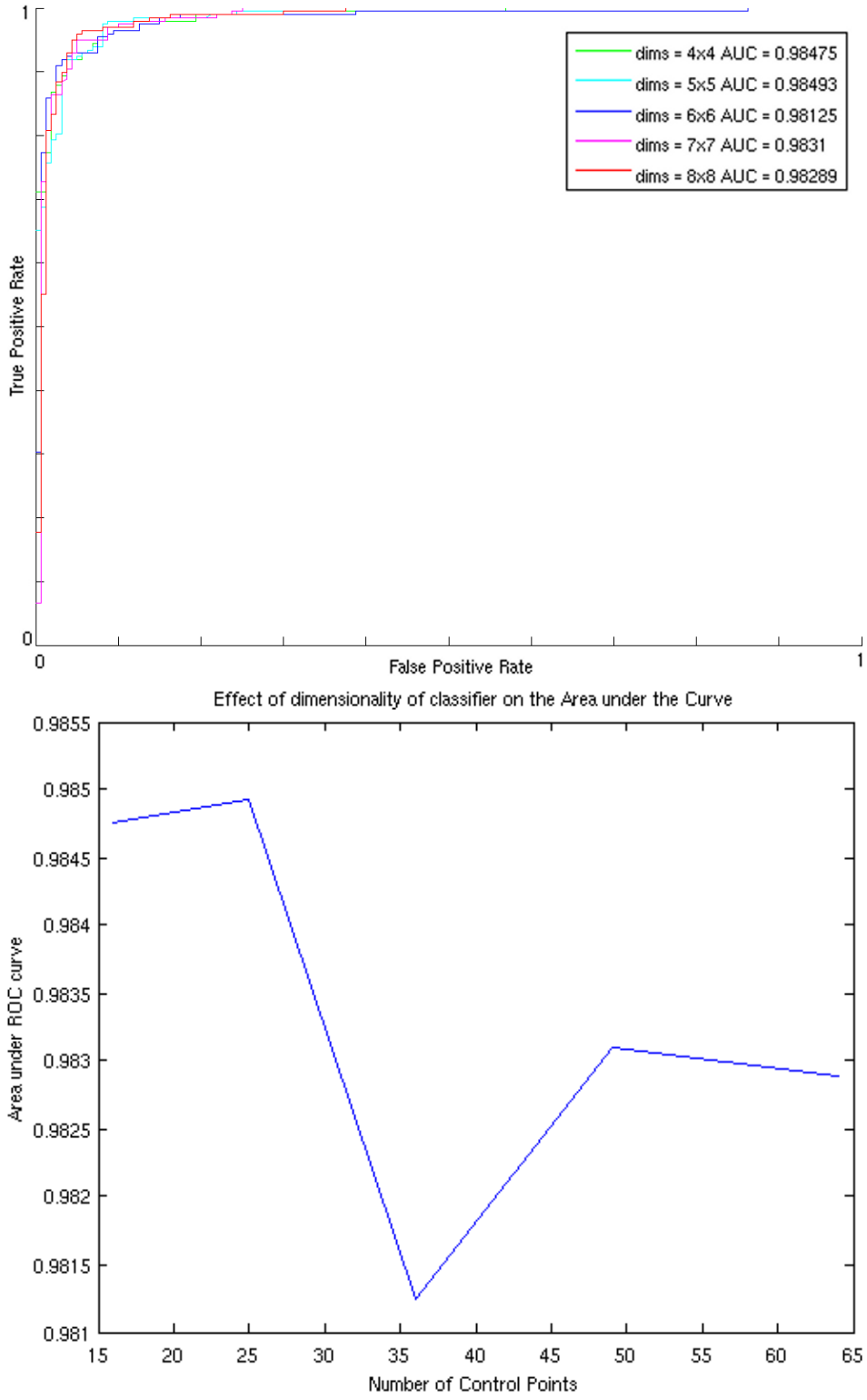


Figure 4.10. Effect of changing the dimensionality of classifier. **Top:** ROC plots for digits 2 and 5 changing dimensionality of classifier. **Bottom:** Effect of changing the number of control points on Area under the ROC curve.

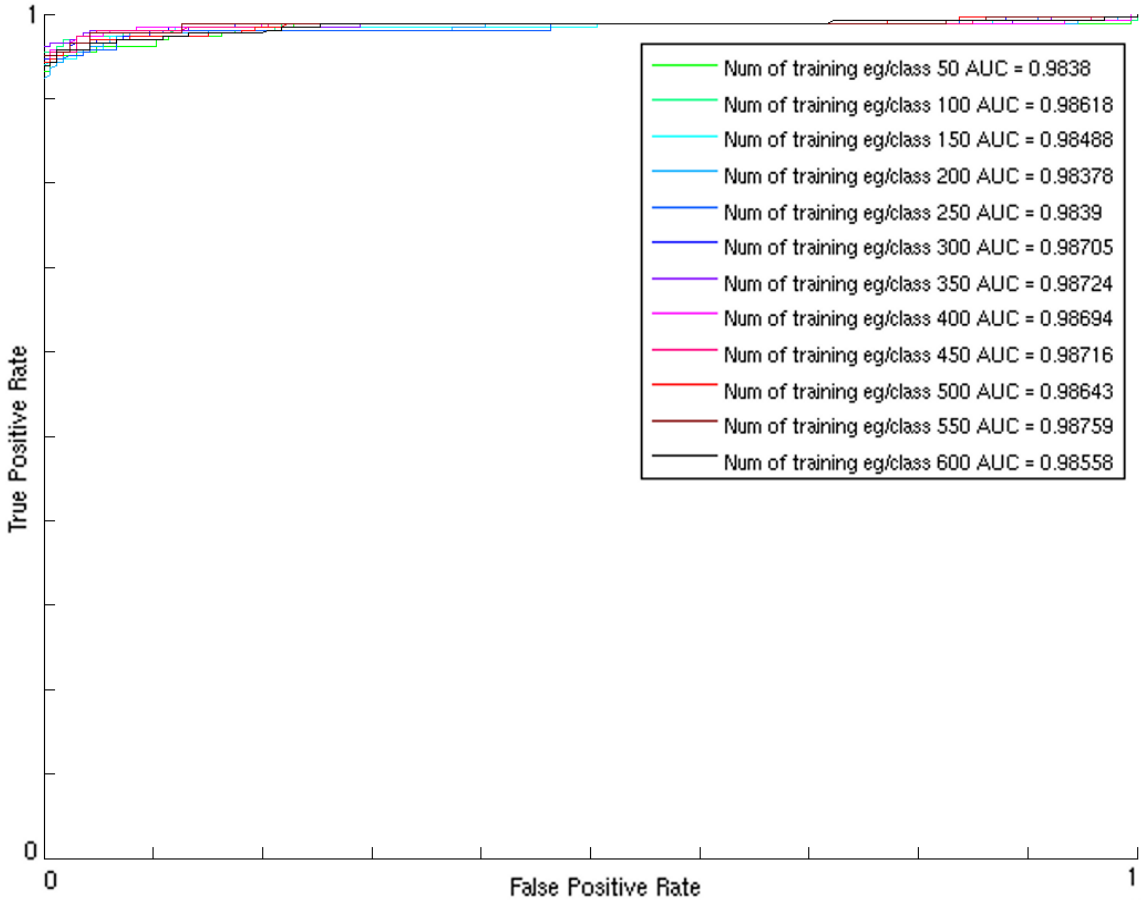


Figure 4.11. Changing the number of training examples for the binary classifier between digits 1 and 3.

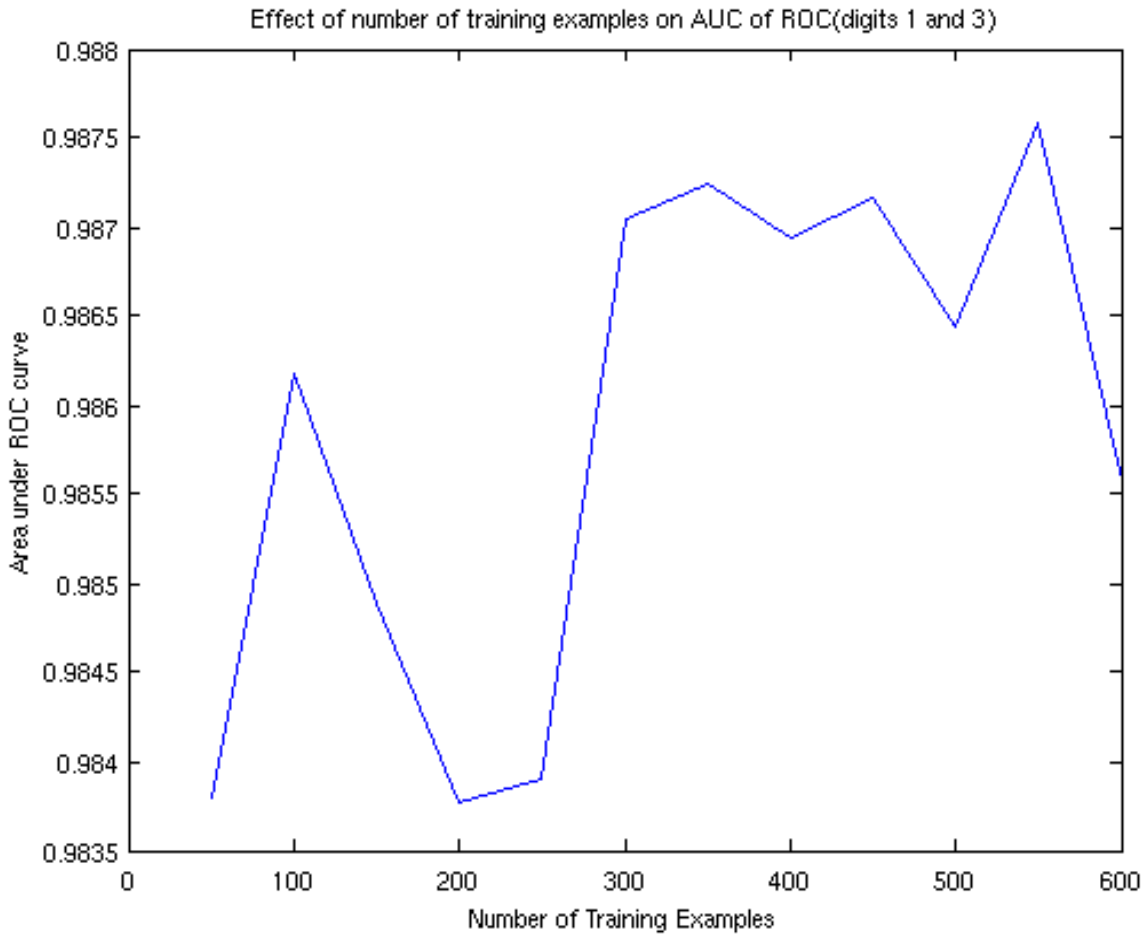


Figure 4.12. Area under ROC curve as the number of training samples is changed.

error tends to decrease with an increase in training samples. This is because we do not know the underlying distribution of the training data.

CHAPTER 5

MULTICLASS CLASSIFICATION

The handwritten digits dataset has 10 classes. In view of the results obtained for binary classification and the need to distinguish between 10 classes, we were encouraged to move onto multiclass classification using simple extensions of the binary classification metrics discussed in section 4.1.

Note that these multiclass classifiers follow the OVA (One versus All) paradigm of classifiers as discussed in [7] since we compare the similarity criterion of the test image with each class and then find the one to which it is closest. Let us first start with a discussion of the classification criterion extensions.

5.1 Multiclass classification extensions

The three classification criteria which we have discussed for binary classifiers in section 4.1 are extended in order to differentiate between multiple classes as follows:

1. **Mahalanobis distance**: In binary classification, we have restricted ourselves to comparing the distance between 2 classes. If we extend this comparison as given in equation 4.4 to all the 10 classes that we have, then we get the classification criterion given in equation 5.1

$$\hat{y} = \arg\min_{l \in \{1,2,\ldots 10\}} M_l(\boldsymbol{\alpha}^l_{test}) \tag{5.1}$$

   The notion behind this classification criterion is that the closer the deformation is to the mean deformation of a certain class, the more likely it is to belong to that class. The notion of the distance that was used for the binary classification has been extended to multiclass classification. Note that this classification criterion is simple and cannot give good results if the clusters overlap.

2. **Magnitude of the momenta vectors**: The same notion of the magnitude of the momenta vectors discussed in equation 4.5 is extended in equation 5.2 to accommodate multiple classes:

$$\hat{y} = \arg\min_{l\in\{1,2,\ldots 10\}} ||\boldsymbol{\alpha}^l_{test}||_{mag} \tag{5.2}$$

$$\text{where } ||\boldsymbol{\alpha}^l_{test}||_{mag} = \sqrt{\sum_i ||\alpha_i||_{L^2}} \tag{5.3}$$

3. **Data matching term**: By far, this metric provides the best classification rates. The metric compares the L^2 norm between the test image and the deformed template image to tell us the difference between the two. The lower the difference, the more likely the test image belongs to the class. The classification can be described as given in equation 5.4.

$$\hat{y} = \arg\min_{l\in\{1,2,\ldots 10\}} ||I^l_{template} \circ \phi(\boldsymbol{\alpha}^l_{test}) - I_{test}||_{L^2} \tag{5.4}$$

As can be seen here, the metric depends upon the quality of the registration, an extension of the rule given in equation 5.4.

5.2 Confusion matrix

The confusion matrix is used to organize and visualize multiclass classifier accuracy. The columns of the matrix represent the class predicted by the classifier, while rows represent the actual class of the instance. The confusion matrix is a square matrix of dimensionality $l \times l$ where l is the total number of classes. The entry x_{ij} in the i^{th} row and j^{th} column represents the number of samples or proportion of samples of class j that have been predicted to be classified as class i. Each multiclass classifier with a specific parameter settings will have one confusion matrix for each test dataset.

It is easy to see if the system confuses two classes using the confusion matrix by looking at the row corresponding to the actual class. When a dataset is unbalanced, the error rate of a classifier is not representative of the true performance of the classifier. This is when the confusion matrix helps us. There are several accuracy measures that have been derived to measure the performance of multiclass classifiers from the confusion matrix as discussed in [12]. Although many measures are proposed, we will be using the simple measure of average error rate across the classes to measure performance of the classifier.

5.3 Multiclass classification using optimally situated control points

First let us plot the confusion matrices for the multiclass classification between a set of 25 optimally situated control points selected using the method described in section 3.2. The three confusion matrices using the classification criteria have been displayed in Figure 5.1. The confusion matrices above have been plotted using the entire training data in the

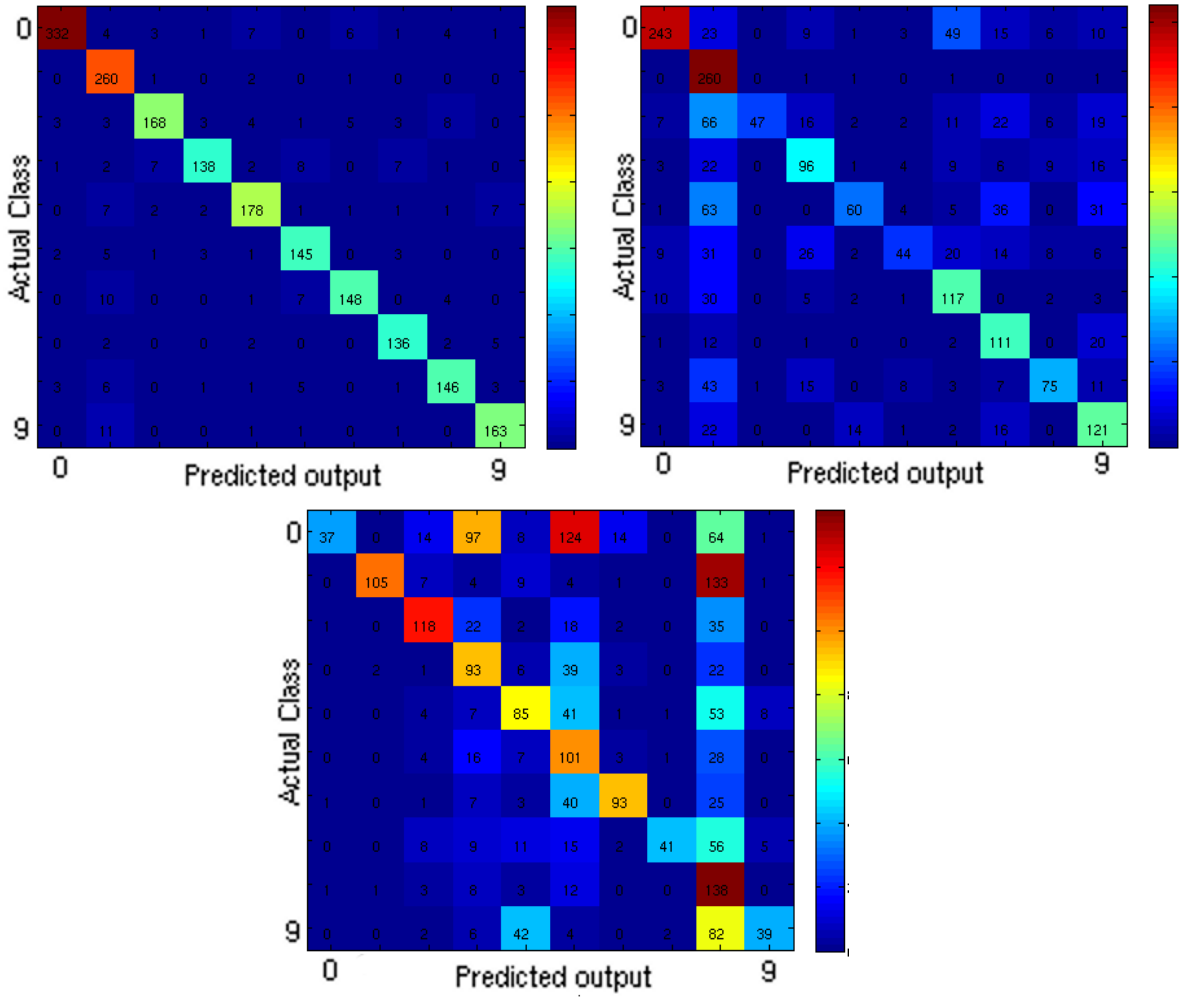


Figure 5.1. Confusion matrices plotted for multiclass classification with 25 control points using $\sigma = 3$, $\gamma_r = 0.25$ with gradient descent on the control point positions using different classification criteria. **Top Left**: Data matching criterion used for classification gives excellent results. Average Error rates are 0.12 **Top Right**: Magnitude of the momenta vectors when used for classification gives average error rate of 0.38. Most digits tend to get confused with the digit 1, 6, 7, and 9. **Bottom**: The Mahalanobis distance does not perform very well with an average error of 0.49. Most digits tend to get confused with the digit 8 as well as with digits 3, 4, and 5.

training phase which are the atlas formation and the entire test data. A cursory inspection of the 3 plots tells us that the data matching criterion of equation 5.4 gives us the best classification results with an average error rate of 12%. The Mahalanobis distance is not a good classification criterion with an average error rate of almost 50% for the current configuration of parameters. This is because the test dataset is incredibly challenging and the deformations from multiple templates tend to look very similar to their respective atlas deformations. Hence, it is more useful to compare the actual deformed image with the test image as is done using the data matching criterion.

The results led to the belief that the low error rate was a result of the deformation descriptor being very low dimensional. The next section will analyze the effect of increasing the dimensionality of the deformation descriptor.

5.4 Using a higher density of control points

In order to analyze the effect of increasing the deformation descriptor dimensionality, we will increase the density of control points. The previous section used 25 regularly distributed control points. Let us have a look at the confusion matrices plotted for the three classification criteria using a grid of $8 \times 8 = 64$ control points. In order to reduce the running time, we have used a slightly smaller training set for construction of the atlas. The results are shown in Figure 5.2.

As can be seen, the average classification error has not changed appreciably by increasing the control point density. The error using the data matching criterion increased from 12% to 12.21%. It is almost stable. The magnitude of momenta criterion has decreased from 38% to 36.71%. Although there is an improvement due to an increase in the resolution in the deformation descriptor, it is not appreciably large. The Mahalanobis distance error changes from 49% to 49.36%, which is not significant.

This set of experiments has led to the conclusion that the data matching criterion yields the best results if we wish to use the velocity field as described in section 1.2 obtained using the registration technique described in Chapters 2 and 3.

To ratify our results, we compare them with the benchmarks discussed in detail in [9]. The dataset used is the ZIP code digits database. From the comparative results given in this paper, we see that the data matching gives results compared to a simple linear classifier. Hence, this is not a good classifier to use all by itself. It should be used in conjunction with other classifier or with higher dimensional images to make the advantages of the system apparent.

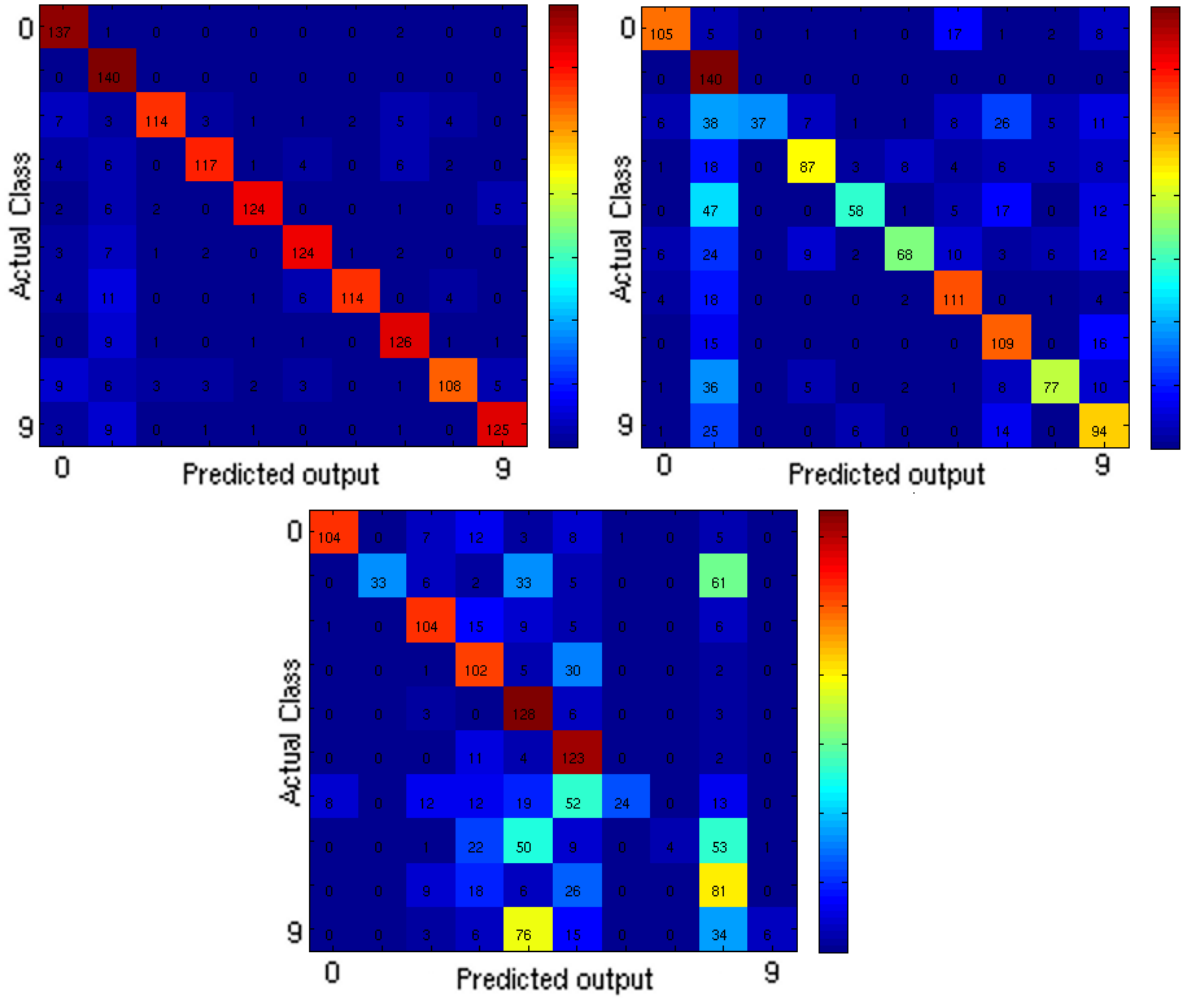


Figure 5.2. Confusion matrices plotted for multiclass classification with 8×8 grid of 64 control points using $\sigma = 3$, $\gamma_r = 0.1$ using different classification criteria. **Top Left**: Data matching criterion has an Average Error rate of 0.1221 **Top Right**: Magnitude of the momenta vectors for classification give average error rate of 0.3671. Confusion with the digit 1,6,7, and 9 occurs frequently. **Bottom**: The Mahalanobis distance has an average error of 0.4936. Most digits tend to get confused with the digit 4 and 8 as well as with 3 and 5.

The error rates obtained in the above set of experiments are high in comparison with results discussed in the LeCunn paper [9]. The above two set of results tell us that in order to use the momenta vectors for performing classification, we need to use a different set of features instead of the image data directly if we want to use the same classification criterion. Otherwise, we may have to optimize all sets of AVA classifiers and build a decision tree-based multiclass classifier. Hence, we have explored the use of one feature in the following section.

5.5 Using the gradient as a feature

One simple attempt towards using a different set of features was using the gradient of the images as a feature instead. In this case, we have formed the atlas using the gradient of the subject images. Also, for classification, instead of using the images directly, we have used the gradient of the images. Thus, in the atlas formation process and the classification process, we have:

$$I_s \mapsto ||\nabla I_s||_{L^2}$$
$$I_{test} \mapsto ||\nabla I_{test}||_{L^2}$$

use the gradient magnitude of the image instead of the image itself

Using the above feature instead of the image itself, Figure 5.3 shows how the multiclass classifier performs.

Average error rates of 49%, 62%, and 58.26% obtained in the above tests tell us that the gradient itself is not a good feature to be used with this method.

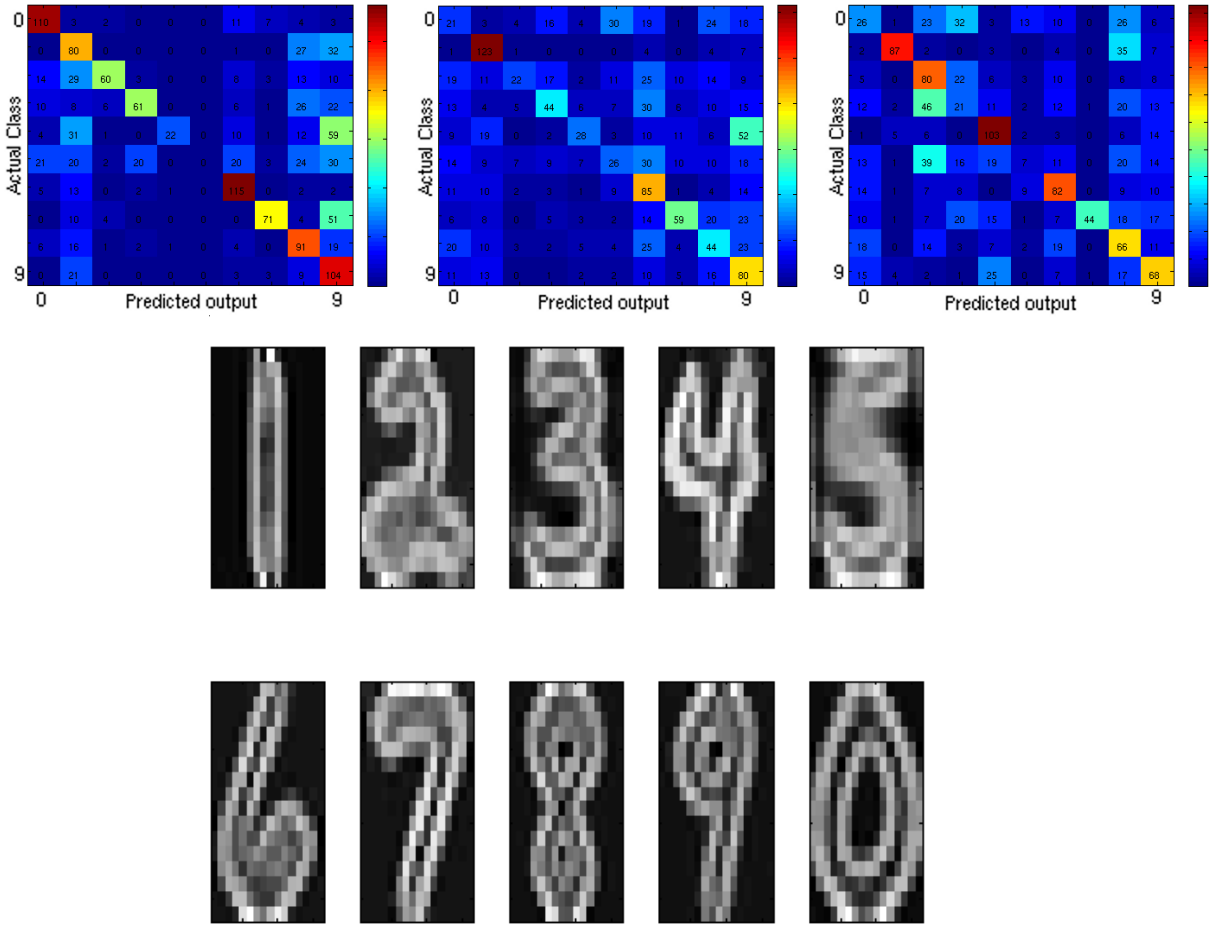


Figure 5.3. Confusion matrices plotted for multiclass classification using gradient of the images as the image feature with 8×8 grid of 64 control points using $\sigma = 3$, $\gamma_r = 0.1$ using different classification criteria. **Left**: Data matching criterion has an average error rate of 0.49 **Middle**: Magnitude of the momenta vectors for classification give average error rate of 0.62. **Right**: The Mahalanobis distance has an average error of 0.5826.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The method of using the velocity field obtained using the registration framework of landmark matching is successful for performing statistics on high-dimensional data. The low-dimensional descriptor at control points improves the results of the classification process. Let us discuss the results obtained along with the implications.

6.1 Image registration and atlas formation

The image registration framework is based on a gradient descent over an objective that balances the smoothness and image match. The gradient descent is stable and is influenced largely by the kernel width of the Gaussian kernel. The optimal kernel width that should be used is equal to the width of the contours in the image. This results in the control points being able to influence the motion of the contours. The contours or level set boundaries are the sections of the image that influence the correspondence or match between images. The time required by the gradient descent to converge increases as the difference between two images increases but is balanced by the breaking ratio stopping criterion. If the gradient descent is not able to influence the objective function value much, the gradient descent is terminated.

The atlas formation using iterative averaging shrinks the template images. This is due to the regularity term as discussed in section 3.1.2. Splatting results in better atlas reconstruction. Also, precomputing an optimal set of landmarks to be used for atlas construction results in better classification results. Using the variance in the atlas formation process as discussed in section 3.2 also helps in reducing the dimensionality and improving the classifier accuracy. The process of atlas construction is time consuming but can be done off-line before the actual classification.

The registration process has a time complexity of $O(NM)$ where N is the number of control points and M is the size of the image in terms of the number of pixels in it. With the relevant parameters settings, adaptive step length, the number of iterations required for convergence is small and constant. Thus, assuming a constant number of

control points, registration has a time complexity linear to the size of the image. Hence, it is computationally efficient.

6.2 Binary classifier

The performance of the binary classifiers was tested using ROC curves. It is found that the performance of the classifiers is optimal for an optimal setting of kernel width σ and regularity trade-off γ_r. This parameter setting is dependent upon the average width of the contours of the data. This can be inferred from the fact that the optimal value of σ is 2 for the binary classifier between digits 1 and 3 while the optimal σ is 3 for the classifier between digits 2 and 5. This is verified not only by the variance of the ROC curves but also by the area under the curve. This tells us that the parameter setting of kernel width should be made so that it is optimal for the data under consideration.

We have also analyzed the effect of varying the dimensionality of the deformation descriptor. It can be seen from section 4.4 and Figure 4.9 that the classifier is accurate for an optimal dimensionality of the deformation descriptor. This extends the results from [3] and proves that the deformation descriptor has an optimal dimensionality. A descriptor with low dimensionality cannot capture all modes of variation of the data while a high dimensional descriptor adds noise artifacts and can bias the classifier towards a certain class.

We also wanted to analyze the effect of the number of training examples on the classifier output. As seen in 4.12, the accuracy of the classifier tends to increase with an increase in the number of training samples. This is the output that we would expect, but it assumes that the test and training samples have been drawn from a uniform random distribution. The trend is somewhat unclear, although it seems to increase due to lack of knowledge of the underlying distribution.

6.3 Multiclass classification

The performance of multiclass classifiers is measured by using confusion matrices and average error rates across the classes. The first set of experiments is performed using an optimally situated set of 25 control points. In the case of the binary classifiers, the Mahalanobis distance criterion yields good results. However, for multiclass extensions of the three metrics used, it is seen in Figure 5.1 that the data matching criterion yields the best results. The data matching criterion has an error rate of 12%. This is due to the fact that as the number of classes increase, the deformations from the test image to multiple class templates look similar. The Mahalanobis distance is not a good metric to use when

the number of classes increases. In our case, this is also due to smaller degrees of variation in the digits dataset.

Increasing the density of control points does not have an appreciable affect on the classifier accuracy. As seen in section 5.4, we can see that the average error rate does not improve much. Figure 6.1 shows an excerpt from [9] that details the error rates obtained for various techniques using the ZIP codes database that we have used.

Comparing to the output of our classifier, we can see that we get results similar to a simple linear classifier. Thus, we have the same computational complexity with far fewer parameters in our system as compared to the linear classifier discussed in [9]. The linear classifier discussed in [9] uses 7850 free parameters. Thus, we have achieved better run times by reducing the dimensionality of the descriptor while keeping the error rate low.

6.4 Future work

The objective of this effort was to quantify the utility of deformation fields as features for tasks such as classification. The technique of using the parametrized optimal deformation field found by a gradient descent on the objective that enforces data matching and smoothness of deformation works well with the given data.

In order to improve the accuracy of the classifier, we will have to incorporate the results of multiple classifiers. This can be done by a weighted sum of the result using ensemble techniques like bagging and boosting. Classifiers using structural information such as ones performing skeletonization of the curves have been shown to yield good results on the digits database [2]. Combining the results of such multiple low-cost classifiers can be more efficient than using expensive classifiers using neural networks [9].

The true potential of this technique can be seen in classification in medical imaging data [3]. Anatomical structures have variations that are locally consistent. Thus, in a normalized dataset, the voxels do not move independently. This fact has been incorporated in the objective function as a smoothness constraint. Also, another advantage of this technique is the massive reduction in the dimensionality of the feature descriptor. Such an improvement in computational cost is not possible using expensive techniques such as neural networks. Thus, the next step is using the technique to answer clinical questions using medical data.

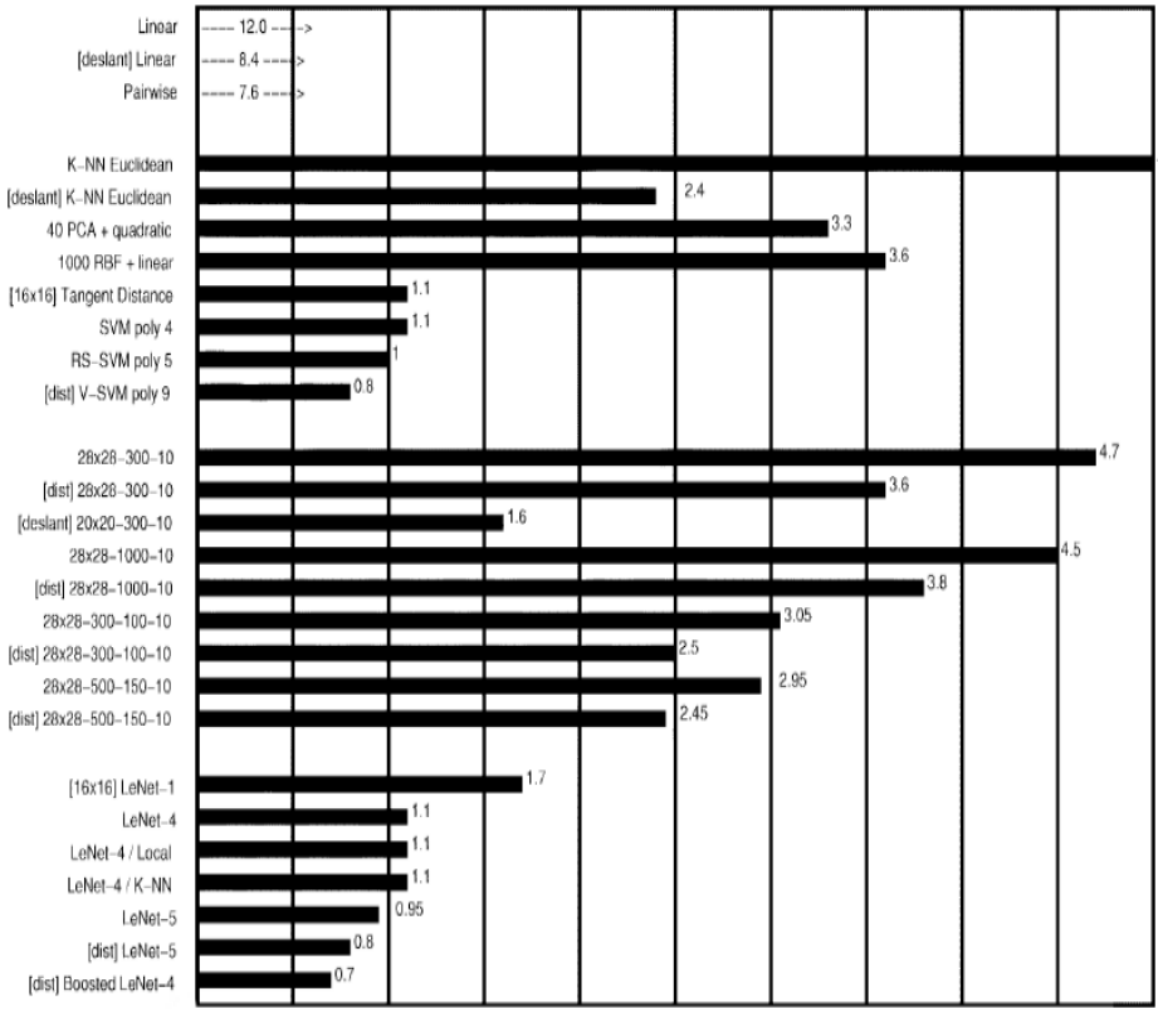


Figure 6.1. Error rate for various classification methods using the ZIP code digits database. Data taken from [9]

REFERENCES

[1] M. F. Beg, M. I. Miller, A. Trouv, and L. Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, International Journal of Computer Vision, 61 (2005), pp. 139–157. 10.1023/B:VISI.0000043755.93987.aa.

[2] S. Behnke, M. Pfister, and R. Rojas, *Recognition of handwritten digits using structural information*, in Neural Networks, 1997., International Conference on, vol. 3, IEEE, 1997, pp. 1391–1396.

[3] S. Durrleman, M. Prastawa, G. Gerig, and S. Joshi, *Optimal data-driven sparse parameterization of diffeomorphisms for population analysis*, (2011), pp. 123–134.

[4] L. C. et al, *Zip Code Normalized handwritten digits*. `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`, 1990. [Online; accessed 19-July-2008].

[5] T. Fawcett, *Roc graphs: Notes and practical considerations for researchers*, (2004).

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer Series in Statistics, Springer, 2009.

[7] H. D. III, *A Course in Machine Learning*, 2012.

[8] S. Joshi and M. Miller, *Landmark matching via large deformation diffeomorphisms*, Image Processing, IEEE Transactions on, 9 (2000), pp. 1357 –1370.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278 –2324.

[10] P. C. Mahalanobis, *On the generalised distance in statistics*, Proceedings of the National Institute of Sciences of India 2, 1 (1936), pp. 49 – 55.

[11] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, 1999.

[12] S. V. Stehman, *Selecting and interpreting measures of thematic classification accuracy*, Remote Sensing of Environment, 62 (1997), pp. 77 – 89.